NASA
Technical
Memorandum

NASA TM - 108457

TRACE CONTAMINANT CONTROL SIMULATION COMPUTER PROGRAM—VERSION 8.1

By J.L. Perry

Structures and Dynamics Laboratory
Science and Engineering Directorate

May 1994

National Aeronautics and
Space Administration

George C. Marshall Space Flight Center

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	May 1994	Technical Memorandum

4. TITLE AND SUBTITLE

Trace Contaminant Control Simulation Computer Program—Version 8.1

5. FUNDING NUMBERS

6. AUTHOR(S)

J.L. Perry

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA TM-108457

11. SUPPLEMENTARY NOTES

Prepared by Structures and Dynamics Laboratory, Science and Engineering Directorate.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified—Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The Trace Contaminant Control Simulation computer program is a tool for assessing the performance of various process technologies for removing trace chemical contamination from a spacecraft cabin atmosphere. Included in the simulation are chemical and physical adsorption by activated charcoal, chemical adsorption by lithium hydroxide, absorption by humidity condensate, and low- and high-temperature catalytic oxidation. Means are provided for simulating regenerable as well as nonregenerable systems. The program provides an overall mass balance of chemical contaminants in a spacecraft cabin given specified generation rates. Removal rates are based on device flow rates specified by the user and calculated removal efficiencies based on cabin concentration and removal technology experimental data. Versions 1.0 through 8.0 are documented in NASA TM–108409. TM–108409 also contains a source file listing for version 8.0. Changes to version 8.0 are documented in this technical memorandum and a source file listing for the modified version, version 8.1, is provided. Detailed descriptions for the computer program subprograms are extracted from TM–108409 and modified as necessary to reflect version 8.1. Version 8.1 supersedes version 8.0. Information on a separate user's guide is available from the author.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Atmospheric Contamination, Contaminants, Life Support Contamination Control, Computer Simulation, Computer Model, Source Files	127
	16. PRICE CODE: NTIS

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unlimited

ACKNOWLEDGMENTS

Version 8.1 of the Trace Contaminant Control Simulation computer program was made possible through the efforts of Mark Leban of Lockheed Missiles and Space Co., who provided the latest information on activated charcoal loading characteristics and modified the code accordingly. Also, he added additional code to account for poisoning of high temperature oxidation catalysts by selected trace contaminants. His expertise and professionalism in making these modifications have been greatly appreciated.

TABLE OF CONTENTS

	Page
INTRODUCTION	1
VERSION 8.0 MODIFICATIONS	1
DETAILED COMPUTER PROGRAM DESCRIPTION	3
Program Editing, Compiling, and Linking	3
Main Program	3
Calculation Loop Subroutines	5
Subroutine CAFILL	5
Subroutine RAFILL	5
Subroutine PCSET	7
Subroutine PRAFIL	8
Subroutine CNRSUB	8
Subroutine MASBAL	9
Subroutine CALCM	9
Subroutine LDIGEN	12
Subroutine PCAVCF	12
Subroutine RINCDD	12
Subroutine REGEN	12
Subroutine MCALC	15
Subroutine PREDCT	15
Subroutine LODEFF	17
Subroutine CONVRG	17
Subroutine SLIOH	19
Contaminant Removal Device Calculation Subroutines	19
Subroutine ACHBD	19
Subroutine RCHBD	24
Subroutine ALIOH	24
Subroutine COOXID	24
Subroutine CATBNR	24
Subroutine CONDHX	26
Data Input and Output Subroutines	28
Subroutine CRIN	28
Subroutine RRIN	28
Subroutine CROUT2	28
Subroutine RROUT2	28
Subroutine CROUT	28
Subroutine RROUT	28
Subroutine DATOUT	28
Subroutine GROUP	28
Subroutine PRFANS	29
Subroutine HEADGS	29

TABLE OF CONTENTS (Continued)

	Page
APPENDIX A – TCCS COMPUTER PROGRAM, VERSION 8.1, FORTRAN CODE LISTING	31
APPENDIX B – TOXIC HAZARD INDEX DESCRIPTION	118
REFERENCES	122

LIST OF ILLUSTRATIONS

Figure	Title	Page
1.	Main program block flow diagram	4
2.	Subroutine PCSET block flow diagram	7
3.	Subroutine CNRSUB block flow diagram	8
4.	Subroutine MASBAL block flow diagram	10
5.	Subroutine CALCM block flow diagram	11
6.	Subroutine RINCDD block flow diagram	13
7.	Subroutine REGEN block flow diagram	14
8.	Subroutine MCALC block flow diagram	16
9.	Subroutine PREDCT block flow diagram	17
10.	Subroutine CONVRG block flow diagram	18
11.	Charcoal saturation and adsorption zone distribution	20
12.	Potential plot for type BD granular activated charcoal	21
13.	Potential plot showing relative humidity effects	22
14.	Adsorption zone length as a function of the potential factor	23
15.	Noble metal CO catalyst performance	25
16.	Absorption of contaminants by humidity condensate	27
17.	Ammonia solubility in water at varying carbon dioxide concentration	27

TECHNICAL MEMORANDUM

TRACE CONTAMINANT CONTROL SIMULATION COMPUTER PROGRAM—VERSION 8.1

INTRODUCTION

The Trace Contaminant Control Simulation (TCCS) computer program development began with the efforts of Olcott which were documented in 1972. Since then, improvements in the user interface and more up-to-date information on activated charcoal loading characteristics and high temperature oxidation catalysts have become available. The progression of the program from version 1.0 through 8.0 and acknowledgments for its development are documented by reference 1. The descriptive material on the computer program subroutines has been extracted in its entirety from reference 1; however, some modifications have been made where necessary.

VERSION 8.0 MODIFICATIONS

The TCCS computer program version 8.0 which is documented in reference 1 has been modified as part of the International Space Station (ISS) design process. These modifications were made initially by personnel at Lockheed Missiles and Space Co. in Sunnyvale, CA, as part of the design activities for the ISS trace contaminant control subassembly. During these activities, new information on activated charcoal loading capacity, influence of relative humidity on charcoal loading capacity, and preliminary information on poisoning characteristics of high temperature oxidation catalysts was obtained. This information was assessed by both Lockheed and NASA Marshall Space Flight Center personnel and integrated into the TCCS computer program as version 8.1. These modifications were officially accepted by NASA in March 1994. Specific details on the modifications can be found in reference 2. A listing of the source files for version 8.1 is contained in appendix A. Version 8.1 supersedes all other versions of the program.

A brief summary of changes to version 8.0 which have resulted in the new version 8.1 are the following:

1. The main program, MAIN.FOR, was modified in the following ways:

   (a) Matrix TT was increased from 300 by 7 to 750 by 7 to accommodate larger time-dependent data files

   (b) Code which sets the time increment size was moved to precede the code which checks for changes in the basic time increment

   (c) The code was modified to accept changes in the basic time increment size as long as the change occurs between the beginning and end of the current time step rather than at the beginning of the time step only

(d) Changes to the cabin volume now result in a recalculation of all contaminant concentrations

(e) Mission duration output formats have been changed to accommodate number sizes up to five digits.

2. Testing of Barnebey-Sutcliffe types AC and 3032 activated carbon with and without 10 weight percent phosphoric acid impregnation has resulted in new charcoal capacity equations. The programs ACHBD.FOR and RCHBD.FOR were modified in the following ways:

(a) Capacity for water soluble contaminants, those with a Henry's Law constant between 0 and 5, was determined to be a function of the adsorption potential factor, A, only and not a function of relative humidity. Activated carbon capacity for this case is expressed by the following equations:

$$q = 2.1e^{-0.31A} \quad \text{for } A > 8 \ , \tag{1}$$

$$q = 0.5 - 0.0405\,A \quad \text{for } A \leq 8 \ . \tag{2}$$

(b) Insoluble contaminants were found to be a function of adsorption potential factor, A, and relative humidity, H_R. Activated carbon capacity for this case is expressed by the following equations:

$$q = (-1.28\times10^{-6})H_R^2 - (2.64\times10^{-3})H_R + 0.5 + [(1.12\times10^{-6})H_R^2 + (2.08\times10^{-4})H_R - 0.0405]A \quad \text{for } A \leq 8 \ , \tag{3}$$

$$q = [(-9.6\times10^{-5})H_R^2 - (1.88\times10^{-2})H_R - 2.11]e^{-0.31A} \quad \text{for } A > 8;\ H_R \leq 50 \text{ percent} \ , \tag{4}$$

$$q = [(9.6\times10^{-5})H_R^2 - (1.88\times10^{-2})H_R - 2.11]e^{-(0.25+0.0012H_R)A} \quad \text{for } A > 8;\ H_R > 50 \text{ percent} \ . \tag{5}$$

(c) The programs were modified to read cabin percent relative humidity from the device definition matrix, DD, row 1 column 14

(d) Carbon chemisorption capacity for ammonia at its spacecraft maximum allowable concentration (SMAC) was changed to 0.0061 grams of ammonia per gram of carbon to reflect the latest phosphoric acid impregnated charcoal performance data. The previous number was based on theoretical estimates rather than experimental results.

3. Subprogram CATBNR.FOR was modified to reflect poisoning of 0.5-percent palladium on alumina catalyst by halocarbons and sulfide compounds. The result of this poisoning is a decrease in the removal efficiency, η, for methane. The efficiency calculation was modified to account for the total mass of halocarbon and sulfide compounds, P, in milligrams by using the following equations:

$$\eta = 97.506\times10^{-0.00010507\,P} \quad \text{for } P \leq 5{,}500 \text{ mg} \ , \tag{6}$$

$$\eta = 31.453 - (1.151\times10^{-3})P + (1.9045\times10^{-8})P^2 - (1.0389\times10^{-13})P^3 \quad \text{for } P > 5{,}500 \text{ mg} \ . \tag{7}$$

4. The following changes were made to subprogram CNRSUB.FOR:

   (a) Modifications were made to allow transferring the relative humidity value to ACHBD.FOR and RCHBD.FOR

   (b) Partial catalytic oxidizer efficiency restoration for methane removal is set to coincide with axial and radial charcoal bed regeneration

   (c) Cumulative masses of halocarbons and sulfides removed by the catalytic oxidizer are transferred to CATBNR.FOR. The cumulative mass is reinitialized at charcoal bed regeneration.

   (d) Code was added to allow for reinitialization of methane oxidation efficiency in CATBNR.FOR if an upstream adsorption device is regenerated.

DETAILED COMPUTER PROGRAM DESCRIPTION

A detailed description of the TCCS computer program source files is provided to acquaint the user with the main program and each significant subroutine. Flow charts of these routines are provided and discussion of the theoretical basis for some routines is provided where appropriate. A listing of the program source files is provided in appendix A. This description is paraphrased from a description produced by Lockheed Missiles and Space Co, Inc., under contract NAS8-36406. This work served as the primary reference for this section, and all block flow diagrams were adapted from this document.[3]

Program Editing, Compiling, and Linking

This program was edited, compiled, and linked using the Ryan-McFarland RM/FORTRAN™ version 2.42 which include the RM/FORTE™ project manager. This FORTRAN compiler is recommended for making changes to the source files.

Main Program

The main program, MAIN, is a simple program with no branching and two loops. A flow diagram is shown in figure 1. Each subroutine required for the particular program run is called during each pass of the main calculation loop until the end of the simulation.

Subroutines CAFILL and RAFILL, which write zeros into all the calculation matrices are called initially to initialize each calculation matrix. Next, CRIN and PRIN are called to read the contaminant, device definition, and time-dependent input data into matrices NN, CDI, DD, and TT. The input data are printed line by line, if desired, by calling subroutines CROUT2 and RROUT2. All initial variables such as time increment beginning time, time increment ending time, and the increment counter are zeroed.

The precalculation setup routine, PCSET, is called next. This routine calculates the initial removal efficiency for each removal device, the equilibrium cabin concentration, and the final cabin concentration for a cabin concentration of 1×10^{-20} mg/m^3 for all contaminants. Intermediate and final calculation results are stored in matrices CC and DD.

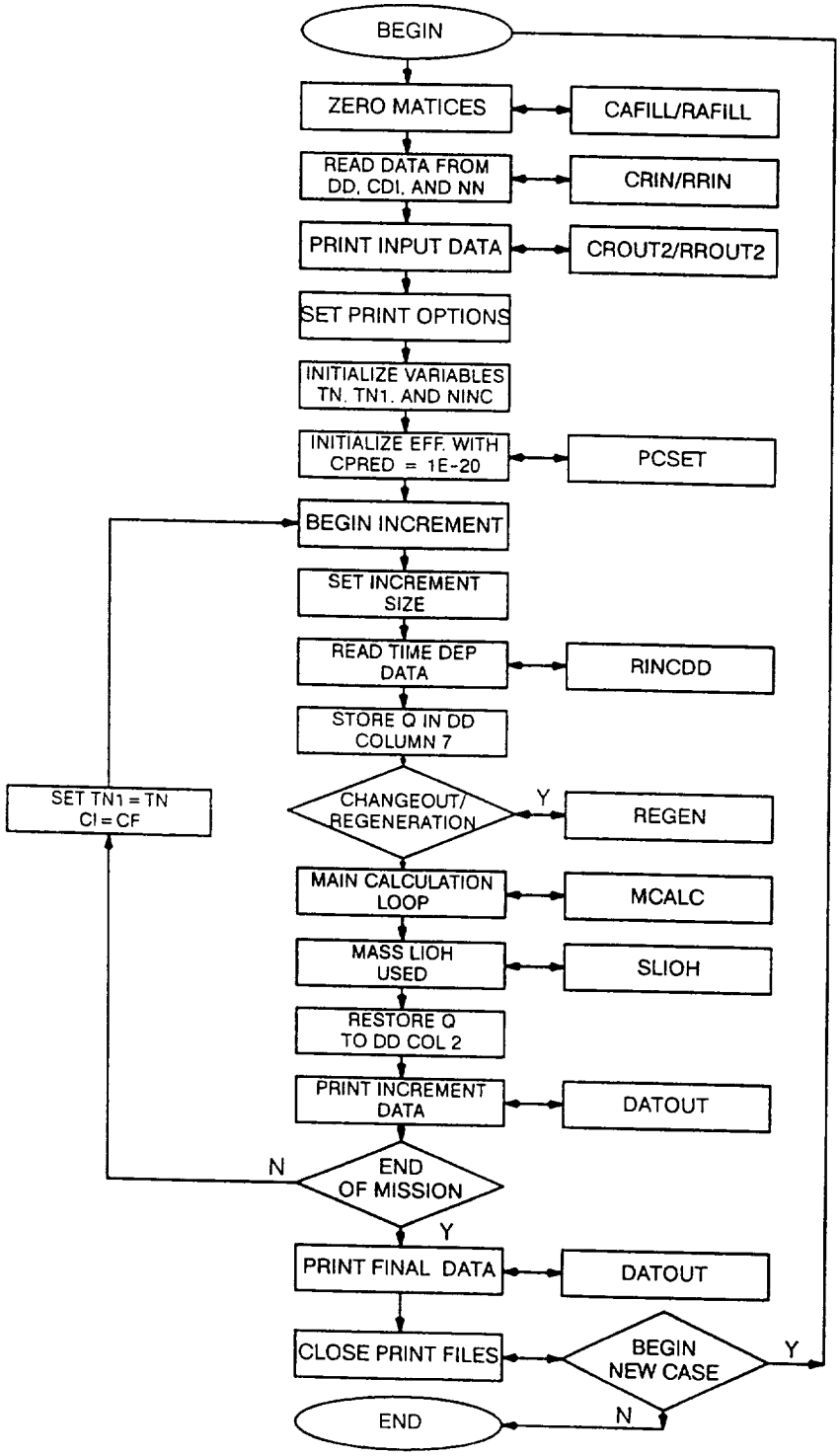


Figure 1. Main program block flow diagram.

The calculation loop is now entered, and the iterative process of determining the cabin concentration for each contaminant at the end of a time increment is begun. Since cabin concentration is a function of the contaminant mass removed and the contaminant mass removed is a function of the cabin concentration, it is important that the same value be used in the mass balance and removal efficiency

calculation routines. A solution is achieved by assuming an increment concentration, calculating an average contaminant concentration, and comparing the two concentrations. This procedure continues until the assumed and calculated concentrations are equal.

Calculation for each time increment is initiated by increasing the increment counter, setting the increment size, and listing the increment number, starting time, and ending time to the computer screen. Subroutine RINCDD is called to read time-dependent data from matrix TT at this time, and subroutine REGEN is called to check for regeneration of any devices during the time increment and to set the adsorbed contaminant masses and device flow rates to zero if necessary. The main calculation loop subroutine, MCALC, is called next to calculate the removal efficiencies, average calculated concentration, and final concentration for each contaminant based on the sum of the mass removed during the previous time increment. Subroutine SLIOH then calculates the amount of lithium hydroxide (LiOH) used during the time increment if a LiOH bed is specified in the device definition file. Next, the original device flow rate is restored for any device that was being regenerated during the increment. Subroutine DATOUT is called to print the calculated data at the end of a time increment, if necessary, to both the standard formatted and plot data output devices. The simulation and mission duration times are then compared to determine whether the mission simulation has ended. If the mission simulation has not ended, another pass through the calculation loop begins by setting the new increment beginning time and initial cabin concentration equal to the previous increment ending time and concentration. If the mission simulation has ended, subroutine DATOUT is called to write the final answers to the appropriate output devices as specified by the user. The output files are closed, and the program loops to the beginning to begin another run if the user wishes. If the user has no other runs to make, the program execution is terminated, otherwise, the calculation matrices are zeroed and new input data is supplied to the program for the next run.

Brief descriptions for each major TCCS computer program subroutine are provided in the order that they are called by MAIN. Table 1 lists the subroutines as they are called and provides a brief description each subroutine's purpose. Block flow diagrams are provided for the most significant subroutines.

Calculation Loop Subroutines

The following subroutines comprise the principal calculation framework for the TCCS Computer Program.

Subroutine CAFILL

The subroutine CAFILL is called by MAIN and fills the matrix NN with blanks. Matrix NN contains the contaminant names during the simulation run.

Subroutine RAFILL

The subroutine RAFILL is called by MAIN and fills the matrices CC, TT, CDI, and DD with zeros. Matrices CDI and CC contains contaminant input and calculation data, matrix DD contains device calculation data, and matrix TT contains time-dependent data. This routine is used at the beginning of a computer simulation to initialize these matrices in the event a previous run has been made.

Table 1. TCCS computer program subroutine listing and description.

Subrouting Level					Description
1	2	3	4	5	
MAIN					MAIN PROGRAM
	CAFILL				ZERO MATRIX NN
	RAFILL				ZERO MATRICES CDI, CC, DD, TT
	CRIN				INPUT DATA INTO NN AND CDI
	RRIN				INPUT FROM FILE TO DD AND TT
	CROUT2				PRINT DATA FROM NN AND CDI
	RROUT2				PRINT DATA FROM DD AND TT
	PCSET				PRECALCULATION SETUP FOR ALL CONT
		PRAFIL			ZERO MATRIX DD COLUMNS 17–21
		CNRSUB			CALCULATE REMOVAL EFFICIENCY
			ACHBD		AXIAL CHARCOAL BED EFFICIENCY
			RCHBD		RADIAL CHARCOAL BED EFFICIENCY
			ALIOH		AXIAL LiOH BED EFFICIENCY
			COOXID		CO OXIDIZER EFFICIENCY
			CATBNR		CATALYTIC OXIDIZER EFFICIENCY
			CONDHX		CONDENSATE EFFICIENCY
		MASBAL			CONTAMINANT MATERIAL BALANCE
			CALCM		SUM OF MASS REMOVED BY DEVICES
			LDIGEN		LOAD GENERATION INTO DD COL 19
			PCAVCF		CALCULATE FINAL AND AVERAGE CONC
	RINCDD				READ INCREMENT DEPENDENT DATA
	REGEN				CALCULATE REGENERATION/CHANGEOUT
	MCALC				MAIN CALCULATION ROUTINE
		PREDCT			CALCULATE PREDICTED AVERAGE CONC
			PRAFIL		ZERO MATRIX DD COLUMNS 17–21
			LODEFF		PUT LAST INCREMENT EFFICIENCY IN DD
			MASBAL		MASS BALANCE ROUTINE
				CALCM	SUM OF MASS REMOVED BY DEVICES
				LDIGEN	LOAD GENERATION INTO DD COL 19
				PCAVCF	CALCULATE FINAL AND AVERAGE CONC
		CONVRG			SOLVE FOR NEW REMOVAL EFFICIENCY
			PRAFIL		ZERO MATRIX DD COLUMNS 17–21
			CNRSUB		CALCULATE REMOVAL EFFICIENCIES
				ACHBD	AXIAL CHARCOAL BED EFFICIENCY
				RCHBD	RADIAL CHARCOAL BED EFFICIENCY
				ALIOH	AXIAL LiOH BED EFFICIENCY
				COOXID	CO OXIDIZER EFFICIENCY
				CATBNR	CATALYTIC OXIDIZER EFFICIENCY
				CONDHX	CONDENSATE EFFICIENCY
			MASBAL		CONTAMINANT MATERIAL BALANCE
				CALCM	SUM OF MASS REMOVED BY DEVICES
				LDIGEN	LOAD GENERATION INTO DD COL 19
				PCAVCF	CALCULATE FINAL AND AVERAGE CONC
	SLIOH				CALCULATE LiOH USED IN INCREMENT
	DATOUT				PRINT DATA TO THE SPECIFIED DEVICE(S)
		PRFANS			PRINT CONCENTRATION DATA ANSWERS
			HEADGS		PRINT DATA HEADINGS
		GROUP			PRINT TOXIC HAZARD INDEX ANSWERS

Subroutine PCSET

The subroutine PCSET is the precalculation setup routine. PCSET gets calculations started by assuming an initial cabin concentration before the program enters the time calculation loop. Figure 2 shows a flow diagram of PCSET. PCSET sets the initial time increment ending time to $1/240$ of the basic time increment specified in the device definition input file. Subroutine PRAFIL is then called and columns 17 to 21 are zeroed. These columns are used to store the results of subsequent calculations. CNRSUB is called to calculate each device removal efficiency for an assumed initial contaminant concentration of 1×10^{-20} mg/m^3. Contaminant removal rates and predicted, equilibrium, and final cabin concentrations are calculated by subroutine MASBAL. These calculated values are copied from matrix DD to the calculation matrix, CC, and printed out by subroutines CROUT and RROUT if required.

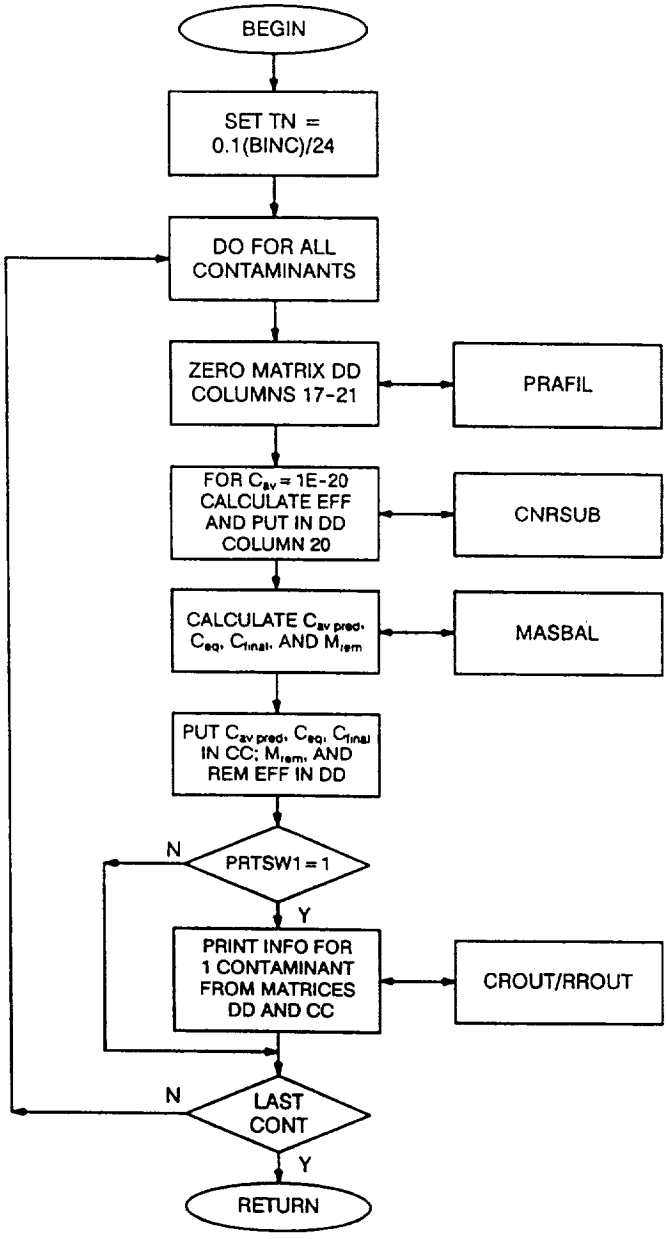


Figure 2. Subroutine PCSET block flow diagram.

Subroutine PRAFIL

Subroutine PRAFIL is called by PCSET and places zeros in matrix DD columns 17 through 21.

Subroutine CNRSUB

The subroutine CNRSUB calculates the removal efficiency of each device for each contaminant in the simulation during every time increment. This calculation is based on the average calculated cabin concentration. Figure 3 shows a block flow diagram for CNRSUB. These calculations are conducted by device type rather than the relative positions of each device with respect to each other.

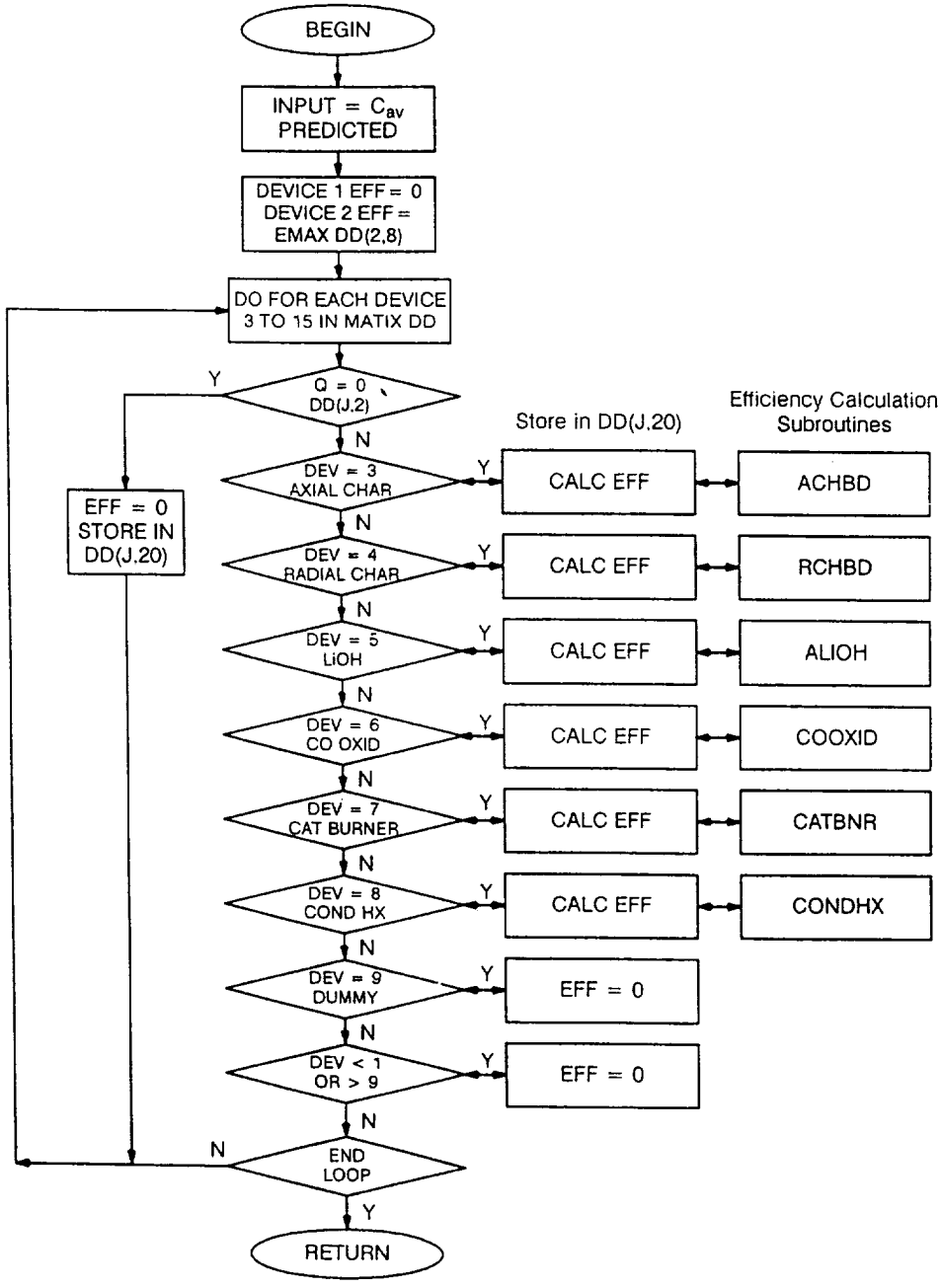


Figure 3. Subroutine CNRSUB block flow diagram.

This routine sets the cabin removal efficiency to zero and the leakage efficiency to the maximum of 1.0. Removal efficiencies for all devices with no flow are also set to zero. The remaining device efficiencies are calculated by calling the subroutines ACHBD, RCHBC, ALIOH, COOXID, CATBNR, and CONDHX. These calculated efficiencies are stored in matrix DD.

Subroutine MASBAL

Using the device efficiencies calculated by CNRSUB, MASBAL determines the mass removed, the calculated cabin concentration, the equilibrium cabin concentration, and final cabin concentration for each contaminant during a time increment. This calculation is conducted for all removal devices in parallel and in series. Figure 4 shows a block flow diagram for MASBAL.

MASBAL uses the mass of contaminant removed and the net mass to the cabin to determine the final cabin concentration for each contaminant. The mass of the contaminant removed is defined as the product of the removal device flow rate, contaminant concentration, and device removal efficiency. The net mass of contaminant to the cabin is defined as the difference between the mass generated and mass removed. At steady state or equilibrium, the mass removed equals the mass generated. The mass generated is the sum of all generation sources which includes the cabin generation rate and the generation rate in each device. The steady-state concentration is defined according to the following equation:

$$C_{ss} = (m_{\text{net-to-cabin}})/(\eta_r \times Q) \ , \tag{8}$$

where $m_{\text{net-to-cabin}}$ is the mass of contaminant, η_r is the overall removal efficiency for all devices, and Q is the atmospheric flow rate through the removal devices.

MASBAL is composed of two parts to determine the steady-state concentration. The first part of MASBAL determines the product of the overall efficiency and flow rate by setting the device generation rates to zero, assuming an arbitrary value for average cabin concentration (100 mg/m^3) and cabin generation rate (50 mg/h), and calling CALCM to determine the sum of mass removed for all the removal devices. The second part of MASBAL evaluates the net mass to the cabin by setting the average cabin concentration equal to zero, restoring the contaminant and device generation rates to the values specified in the contaminant data matrix, and calculating the mass removed using CALCM. The $m_{\text{net-to-cabin}}$ equals the difference between the masses generated in the cabin and removal devices and the mass removed. From these values, C_{ss} is calculated according to equation (8). After calculating the steady-state concentration, the final and average cabin concentrations are calculated by calling PCAVCF, and CALCM is called to calculate the mass removed by the cabin and each device using the average calculated cabin concentration.

Subroutine CALCM

The removal device inlet and outlet concentrations and the total mass removed by the cabin and the specified removal devices is calculated CALCM by using the removal efficiencies, generation rates, and average cabin concentration. This calculation is sequential from one device to another and uses the outlet concentration of an upstream device as the inlet concentration for a downstream device. This calculation requires the device definition input data to be arranged to allow calculations for all upstream devices to be completed before calculations for the downstream devices. Figure 5 shows a block flow diagram for CALCM.

The subroutine sets the cabin and leakage device inlet and outlet concentrations equal to the average cabin concentration. All other devices are tested for zero flow. Devices with zero flow have

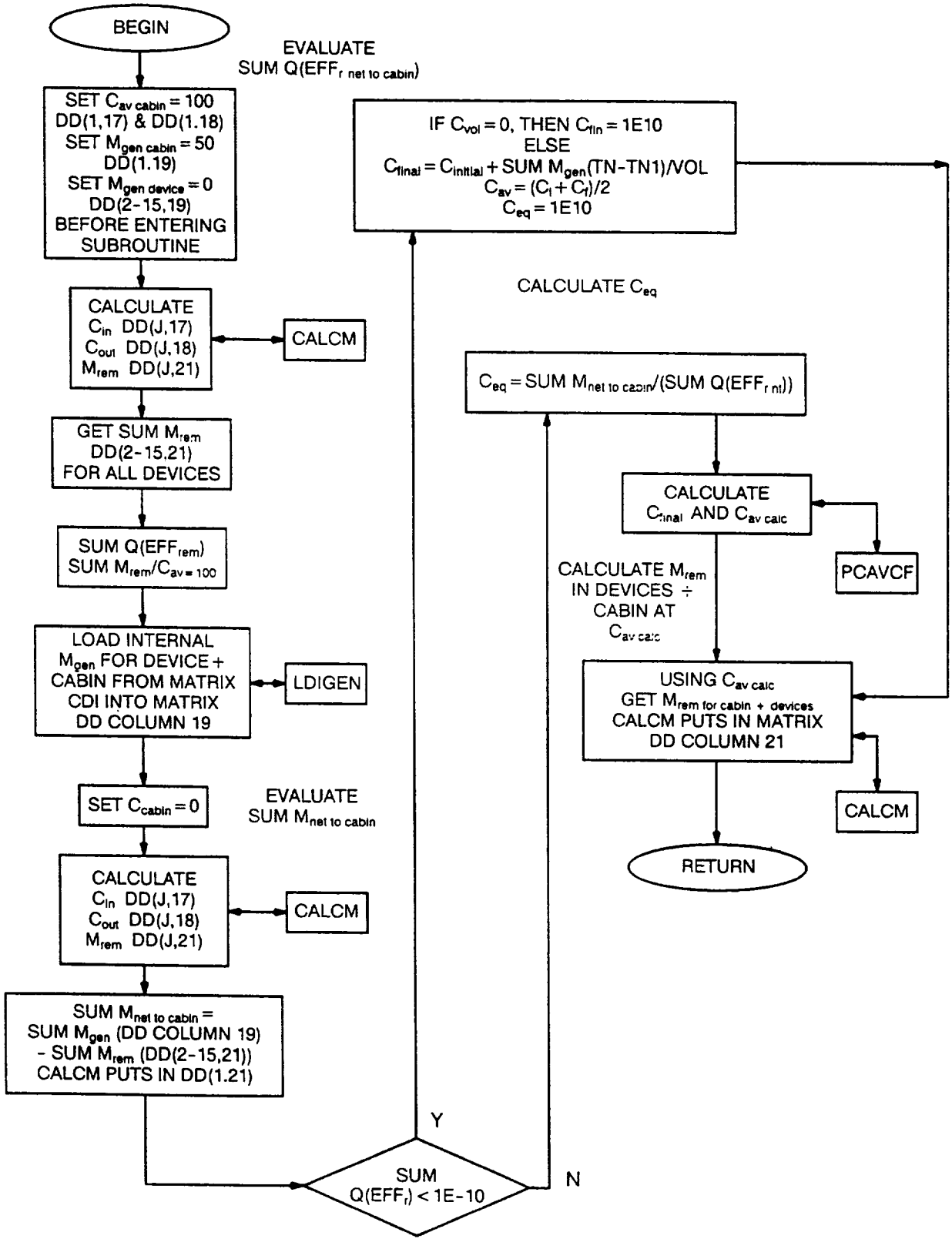


Figure 4. Subroutine MASBAL block flow diagram.

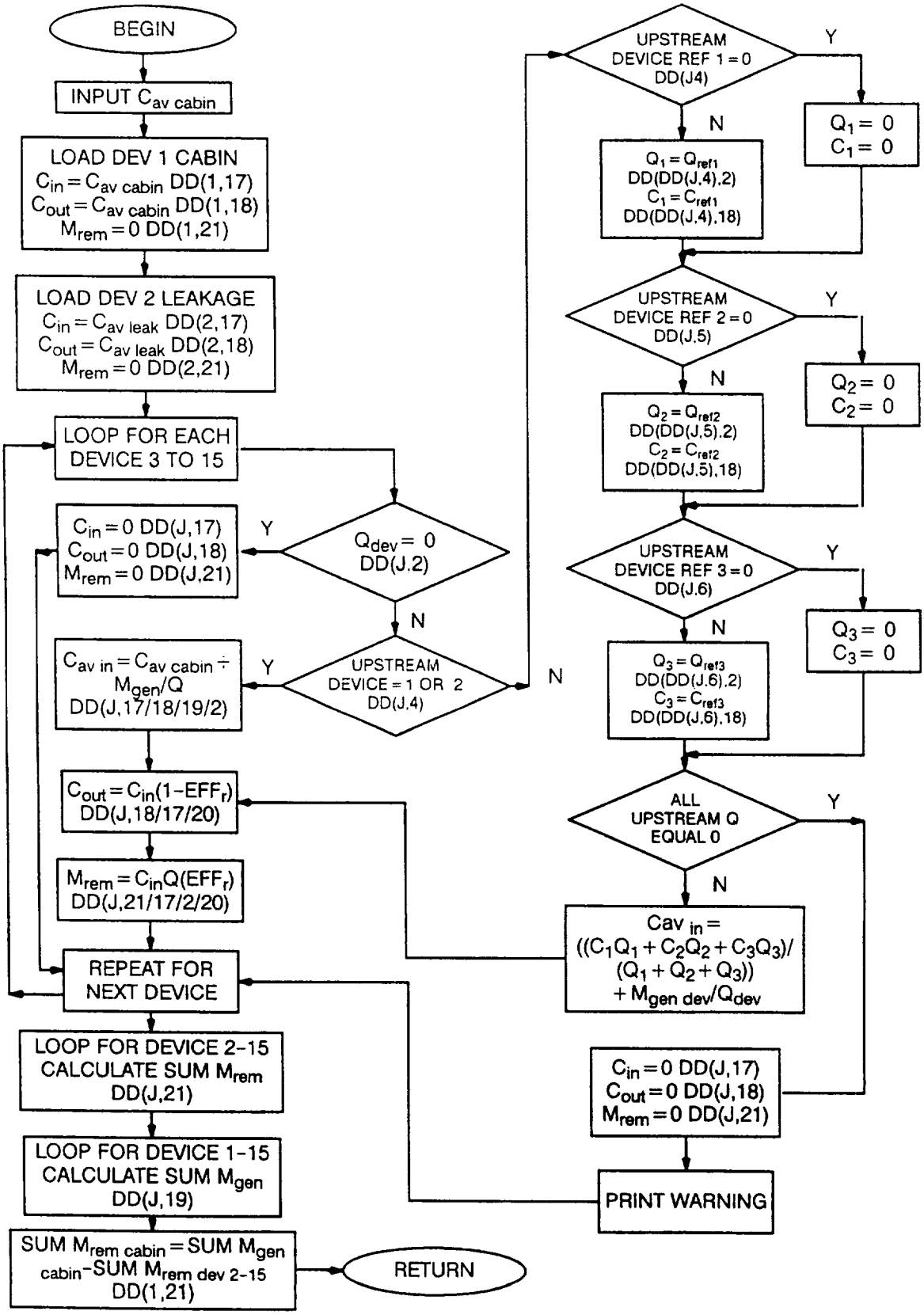


Figure 5. Subroutine CALCM block flow diagram.

their inlet concentration, outlet concentration, and mass removed set equal to zero. Upstream devices for each removal device are identified. If the upstream device type is 1 or 2, the inlet concentration is set equal to the average cabin concentration plus any internal device generation rate divided by the device flow rate. Upstream device types other than 1 or 2 cause the device inlet concentration to be based on the flow rates and outlet concentrations of all the upstream devices.

Outlet concentration and the mass removed by the devices are calculated according to the following equations:

$$C_{out} = C_{in}(1-\eta_r) , \tag{9}$$

$$m_{rem} = C_{in}(Q)(\eta_r) . \tag{10}$$

The inlet concentration for a device with an upstream device is set equal to the outlet concentration for the upstream device. A device with multiple upstream devices requires the mixing of streams with varying concentrations to be considered. For example, the inlet concentration for a device with three upstream devices must be calculated according to the following equation:

$$C_4 = (C_1Q_1+C_2Q_2+C_3Q_3)/Q_4 . \tag{11}$$

The sum of the mass removed and mass generated is calculated by adding the masses removed and masses generated by all the devices. The difference between the sum of the mass generated and the sum of the mass removed gives the mass removed by the cabin.

Subroutine LDIGEN

Subroutine LDIGEN is called by MASBAL to load the generation rates from matrix CDI column 1 and columns 10 through 22 into matrix DD column 19.

Subroutine PCAVCF

Subroutine PCAVCF is called by MASBAL to calculate the increment final and average cabin concentrations for each contaminant.

Subroutine RINCDD

Subroutine RINCDD is used at the beginning of each time increment to input and operate on the time-dependent data. A flow diagram of RINCDD is shown by figure 6. This subroutine checks the time-dependent data to determine whether any changes occur during the current time increment. Variables in matrix TT are identified. If a contaminant generation rate is indicated, the new rate is placed in the calculation matrix, CDI. Likewise, if a change in removal device flow rate or any other device change is indicated, the new information is placed in the appropriate device definition matrix, DD, location.

Subroutine REGEN

Figure 7 shows a block flow diagram for subroutine REGEN. This subroutine determines whether any charcoal or LiOH beds will be regenerated during the current time increment. If regeneration occurs, the mass of contaminants stored in the beds is set equal to zero. Similarly, if the regeneration duration lasts for the entire time increment, the device flow rate is set equal to zero.

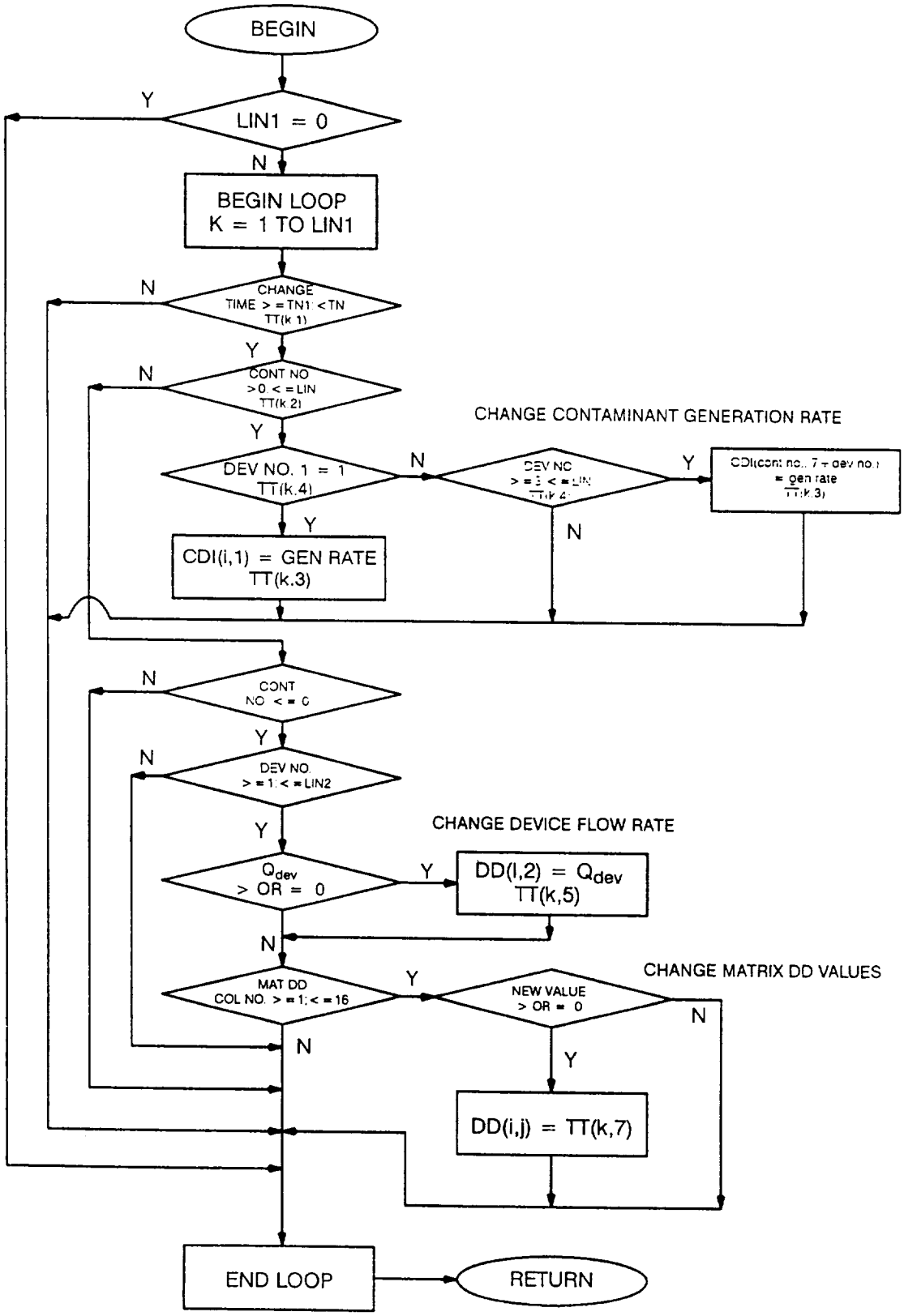


Figure 6. Subroutine RINCDD block flow diagram.

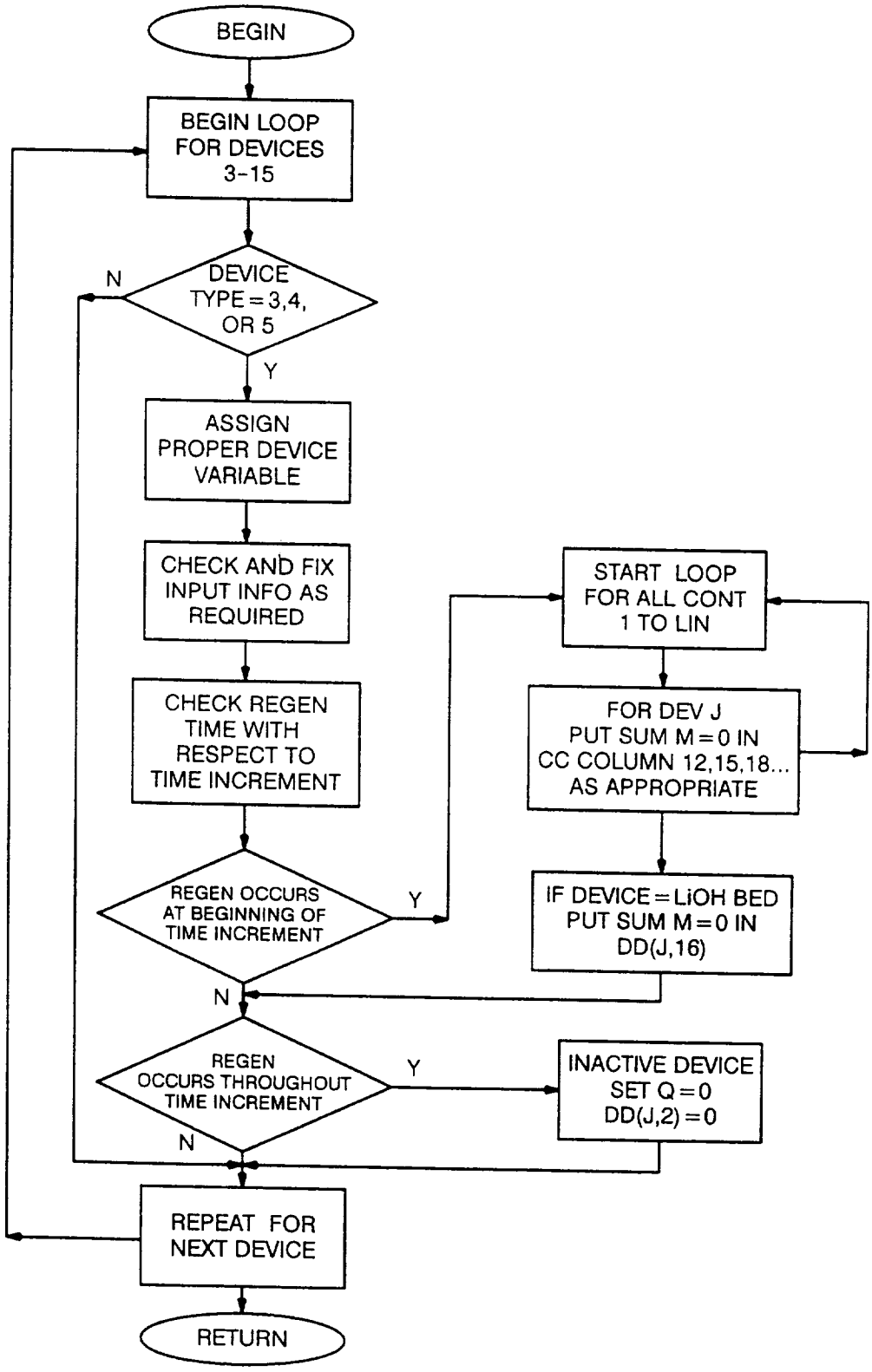


Figure 7. Subroutine REGEN block flow diagram.

The first check conducted by the subroutine is for device type. Only charcoal and LiOH beds may be regenerated. Only regeneration cases which begin at the time increment beginning or regeneration cases which last for one or more complete time increments are treated. Data concerning the regeneration interval, duration, and first regeneration time are obtained from matrix DD. For the LiOH bed regeneration, the duration is set equal to zero since bed changeout is assumed to occur quickly. The initial time, regeneration time, and regeneration duration are then checked to determine whether they are exact multiples of the basic time increment. If they are not, they are rounded to the next lowest multiple of the time increment and a warning is written to the screen.

The next checks conducted by the routine determine whether regeneration occurs at the beginning of a time increment and whether the regeneration lasts for the entire increment. Regeneration for the entire increment causes the program to deactivate this device for that increment by setting the device flow rate equal to zero. Regeneration at the beginning of an increment causes the sum of the mass removed by that device to be set equal to zero. For an LiOH bed, the total mass of LiOH used is also set equal to zero.

Subroutine MCALC

Calculation of the removal efficiency, mass removed, and calculated, equilibrium, and final cabin concentrations for each contaminant and each removal device is controlled by MCALC. These calculations are based on the cumulative mass removed for each contaminant during the previous time increment. Figure 8 shows a block flow diagram for MCALC.

MCALC calls the subroutine PREDCT to calculate the average predicted cabin concentration based on the removal efficiency and the cumulative mass of contaminant removed during the previous time increment and the generation rate during the present time increment. The average predicted concentration is used by the subroutine CONVRG to calculate a new removal efficiency, mass removed, and average calculated, equilibrium, and final cabin concentrations. The predicted and calculated concentrations are compared in CONVRG and recalculated until the difference between them is less than the convergence error specified in the device definition data file, matrix DD. This recalculation and comparison continues for 20 iterations with the full time increment or until the difference is less than the convergence error.

If the convergence error is still exceeded after 20 iterations, another loop with a maximum of 20 iterations is entered which uses one-twentieth the basic time increment for the calculation. This loop ends as soon as the difference between the predicted and calculated values is less than the convergence error or 20 iterations have been completed. If convergence is not attained after this loop, the program writes a warning to the screen indicating that the calculation for the contaminant did not converge. The loop using the one-twentieth time increment is used only for a contaminant that does not converge during the first 20 iterations. This is more efficient than reducing the time increment for all the contaminant calculations.

Subroutine PREDCT

MCALC calls the subroutine PREDCT to calculate the average predicted cabin concentration for each contaminant during each time increment. This calculation is based on the removal efficiency and sum of contaminant mass removed in the previous increment and the generation rate during the present increment. Figure 9 shows a block flow diagram for PREDCT.

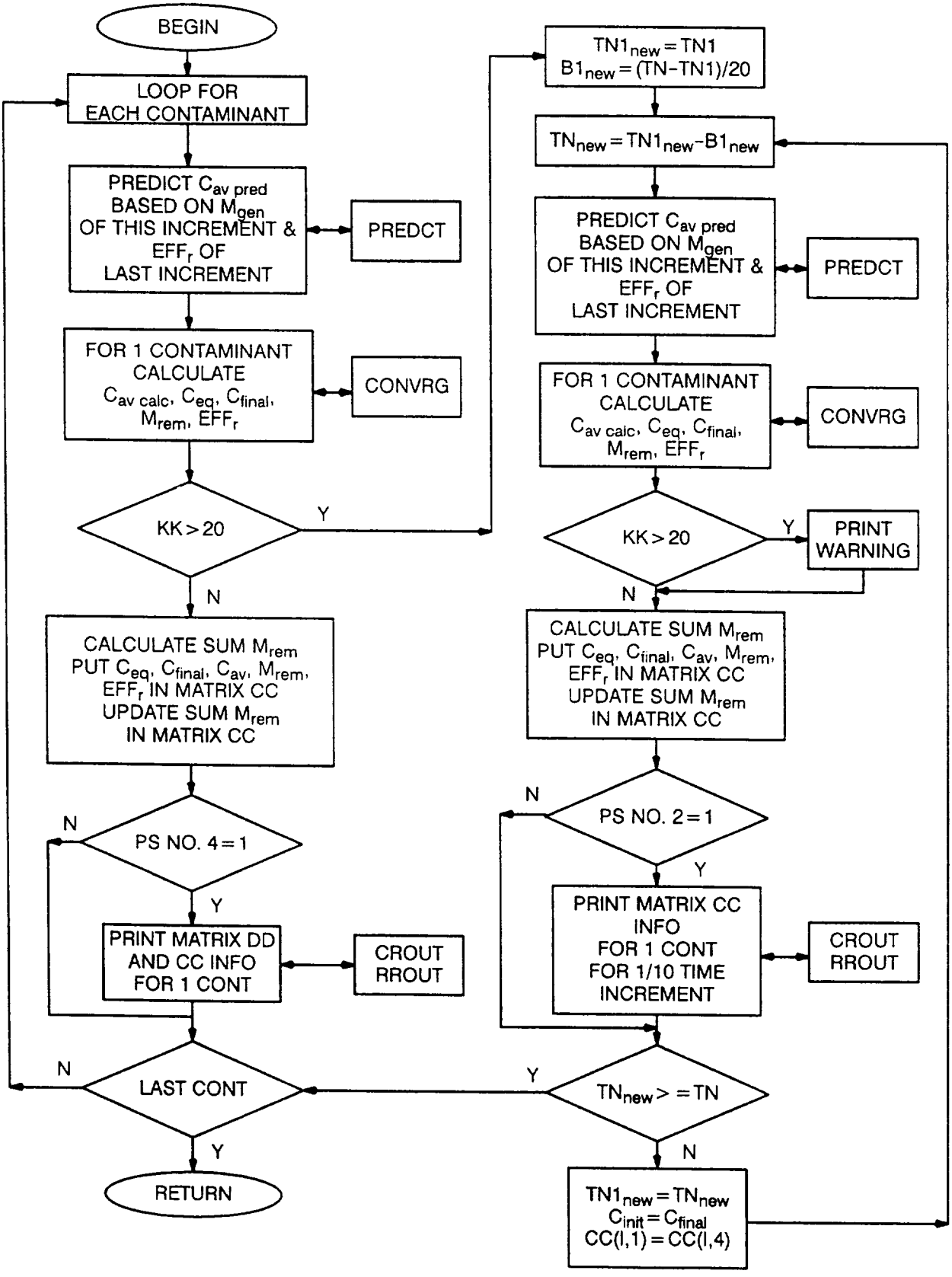


Figure 8. Subroutine MCALC block flow diagram.

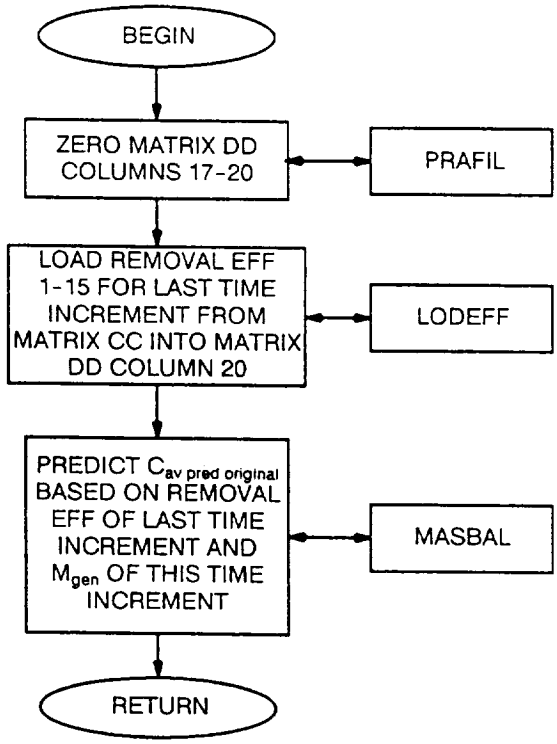


Figure 9. Subroutine PREDCT block flow diagram.

PREDCT calls the subroutine PRAFIL to zero the part of matrix DD required for storing the calculation results. Data from the previous time increment are obtained by LODEFF, and MASBAL is called to calculate the efficiency and concentration.

Subroutine LODEFF

Subroutine LODEFF loads the efficiency calculated in the preceding increment from matrix CC to matrix DD.

Subroutine CONVRG

CONVRG is the main convergence loop subroutine. This subroutine calculates the average cabin concentration and compares it with the predicted cabin concentration for each contaminant during every time increment. Figure 10 shows a block flow diagram for CONVRG.

CNRSUB is called by CONVRG to calculate the removal efficiency for each device using the predicted cabin concentration. Based on this removal efficiency, MASBAL calculates the average, final, and equilibrium cabin concentrations and the mass of contaminant removed by each device. The predicted and calculated cabin concentrations are compared by calculating the absolute value of the difference of the predicted and calculated concentration divided by the predicted concentration and comparing that value to the convergence error. If the absolute value of the comparison is less than the convergence error, convergence has been achieved and the iteration stops for that contaminant. If convergence has not been achieved, a new cabin concentration is calculated using a bisection technique after the first iteration and a Newton-Raphson technique for each additional increment. The loop counter value passes back to MCALC which determines whether convergence has been reached within 20 iterations.

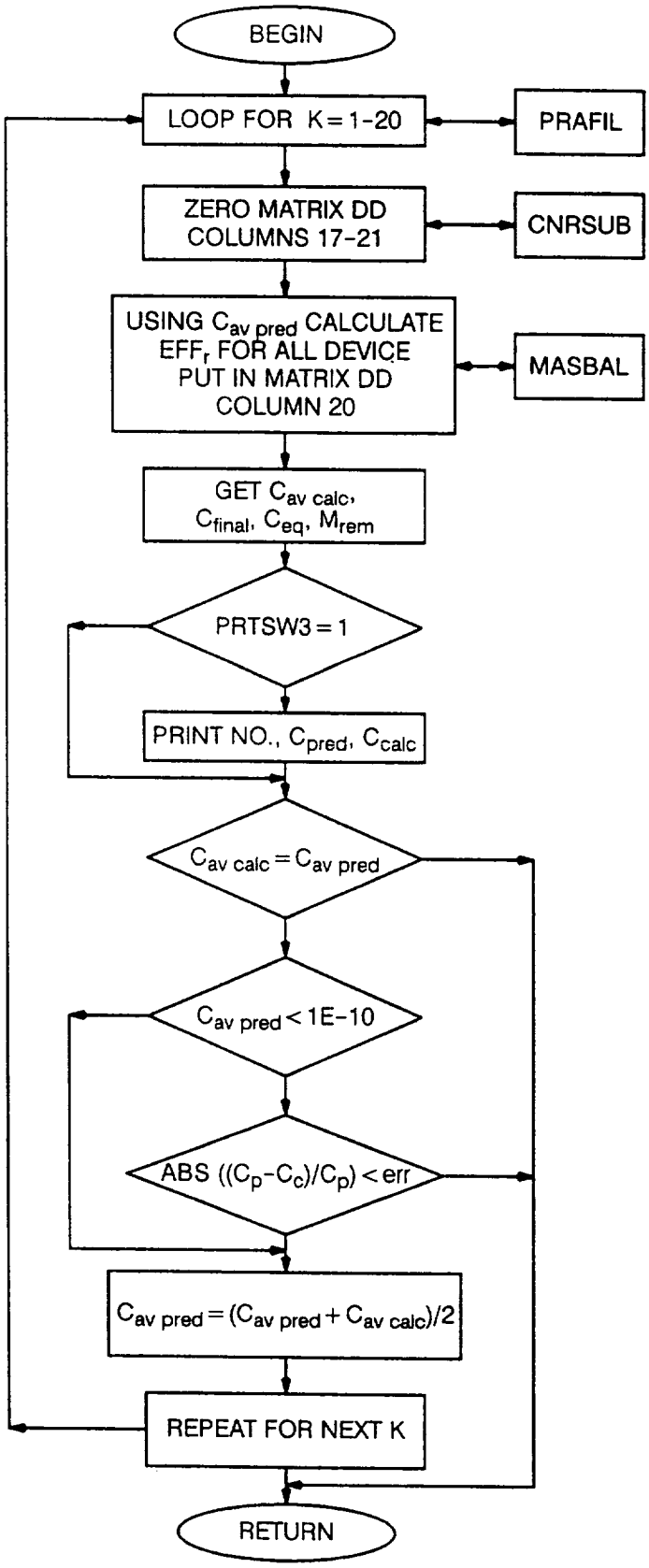


Figure 10. Subroutine CONVRG block flow diagram.

Subroutine SLIOH

SLIOH is the subroutine which calculates the cumulative mass of LiOH used during the simulation run. This calculation is cumulative since the mass of LiOH consumed during the present increment is added to the mass consumed in all the previous increments.

Contaminant Removal Device Calculation Subroutines

The contaminant removal device subroutines are supported by a substantial amount of theoretical and experimental data. A brief description of each subroutine is provided in addition to a discussion of the supporting theory and experimental data.

Subroutine ACHBD

Subroutine ACHBD calculates the removal device efficiency for an axial flow charcoal bed. This routine simulates the physical adsorption of contaminants onto the surface of the charcoal. Specially treated charcoals are also considered which include chemical reaction between the surface treatment and the contaminant in addition to adsorption.

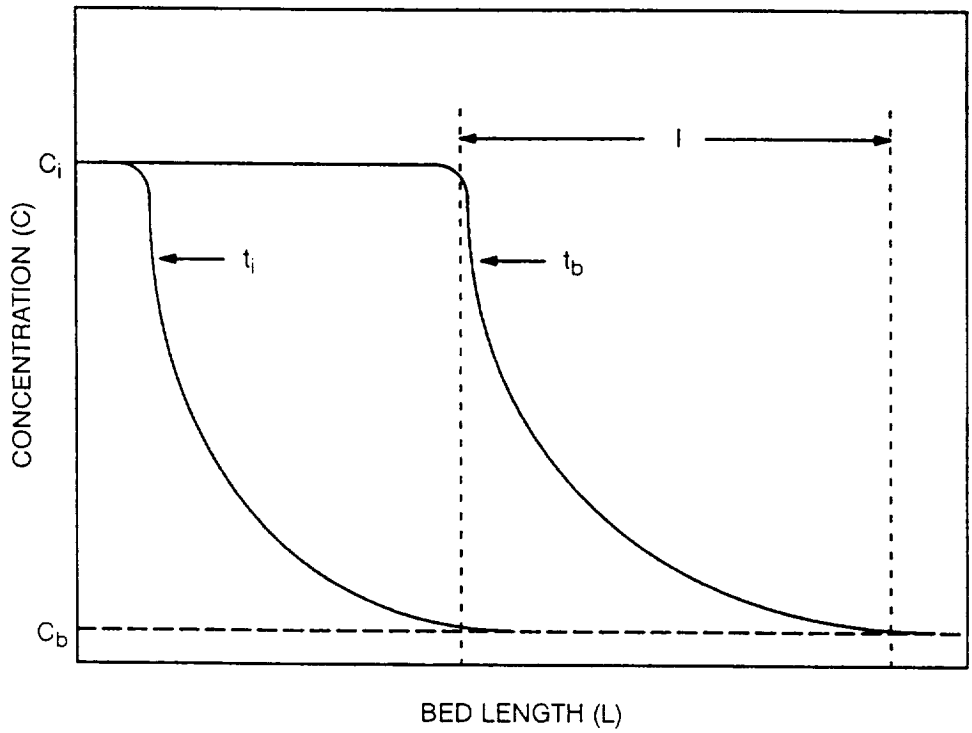
A charcoal bed is composed of two zones during the adsorption process. These zones are designated as the saturated zone and the adsorption zone. All contaminant removal takes place in the adsorption zone. The saturated zone provides no net removal since it is in equilibrium with the vapor phase contaminant composition. Figure 11 illustrates the zones simulated by the program graphically. Physical adsorption is an equilibrium process which depends on variables such as the contaminant vapor pressure, inlet concentration, molar volume, and cabin temperature. Studies conducted by Robell investigated the thermodynamics of adsorption dynamics and developed a correlation between the physical properties of a contaminant and the charcoal saturation capacity. This correlation is based on the Polanyi Potential Theory and the Gibbs equation. From this study, a correlation factor, called the adsorption potential factor was developed. This factor is defined according to the following equation:

$$A = (T/V_m)\log_{10}(p_v/p_c)\,, \tag{12}$$

where T is the cabin temperature in Kelvin, V_m is the contaminant liquid molar volume in cm^3/gram-mole, p_v is the contaminant vapor pressure at the cabin temperature expressed in concentration units of mg/m^3, and p_c is the cabin contaminant partial pressure expressed in concentration units of mg/m^3. This factor was plotted as a function of experimentally determined charcoal saturation capacities to obtain the plot shown by figure 12.[4] The plot in this figure was constructed for Barnebey-Sutcliffe type BD granular activated charcoal. This correlation is not only sensitive to charcoal impregnation and contaminant solubility, but also to relative humidity as shown by figure 13.[5] Additional information may be obtained on charcoal capacity and performance from references 6 and 7. Based on potential plots, empirical equations are obtained which relate the potential factor to the charcoal saturation capacity. The equations used in this program are functions of the adsorption potential factor, contaminant solubility, and cabin relative humidity. Specific equations used are found in subroutine FQI in the ACHBD.FOR listing found in appendix A. The general form of the equations is the following:

$$q = \alpha\, e^{-\beta A}\,, \tag{13}$$

where q is the charcoal saturation capacity in cm^3 of liquid contaminant per gram of charcoal and A is the adsorption potential factor in Kelvin-gmol/cm^3. As new information concerning adsorption capacity



LEGEND

t_i = Bed profile at the time when a bed segment reaches steady state
t_b = Bed profile at the service time when the bed outlet concentratio equals C_b
I = Active adsorption zone length
C_i = Bed inlet concentration
C_b = Bed penetration concentration

Figure 11. Charcoal saturation and adsorption zone distribution.

is obtained, these equations can be modified accordingly. In addition, this technique can be applied to other adsorbent materials to simulate other packing materials besides charcoal.

The adsorption zone length for ninety percent removal is determined from experimental data obtained by Olcott at a 0.0066 m/s (1.3 ft/min) flow rate.[8] This data is plotted in figure 14 and the computer program uses the following equation to calculate the adsorption zone length:

$$L_{ads} = (L_{ads \text{ at } 1.3 \text{ ft/min}})(V/1.3)^{0.8} . \quad (14)$$

The adsorption zone length study conducted by Olcott shows that the adsorption zone length increases with velocity to the 0.8 power as indicated in the equation. The saturated zone length is based on the charcoal capacity at the prevailing cabin conditions and the amount of contaminant already adsorbed. For a given contaminant mass retained in the bed, the saturation zone length equals the mass of contaminant adsorbed divided by the saturation capacity, q. The total bed length minus the saturated zone length equals the adsorption zone length. The adsorption zone length is the length of the bed actually available for contaminant removal. The program calculates the saturation zone length as calculated based on the bed geometry, the amount of contaminant removed by the bed, and the saturation capacity, q.

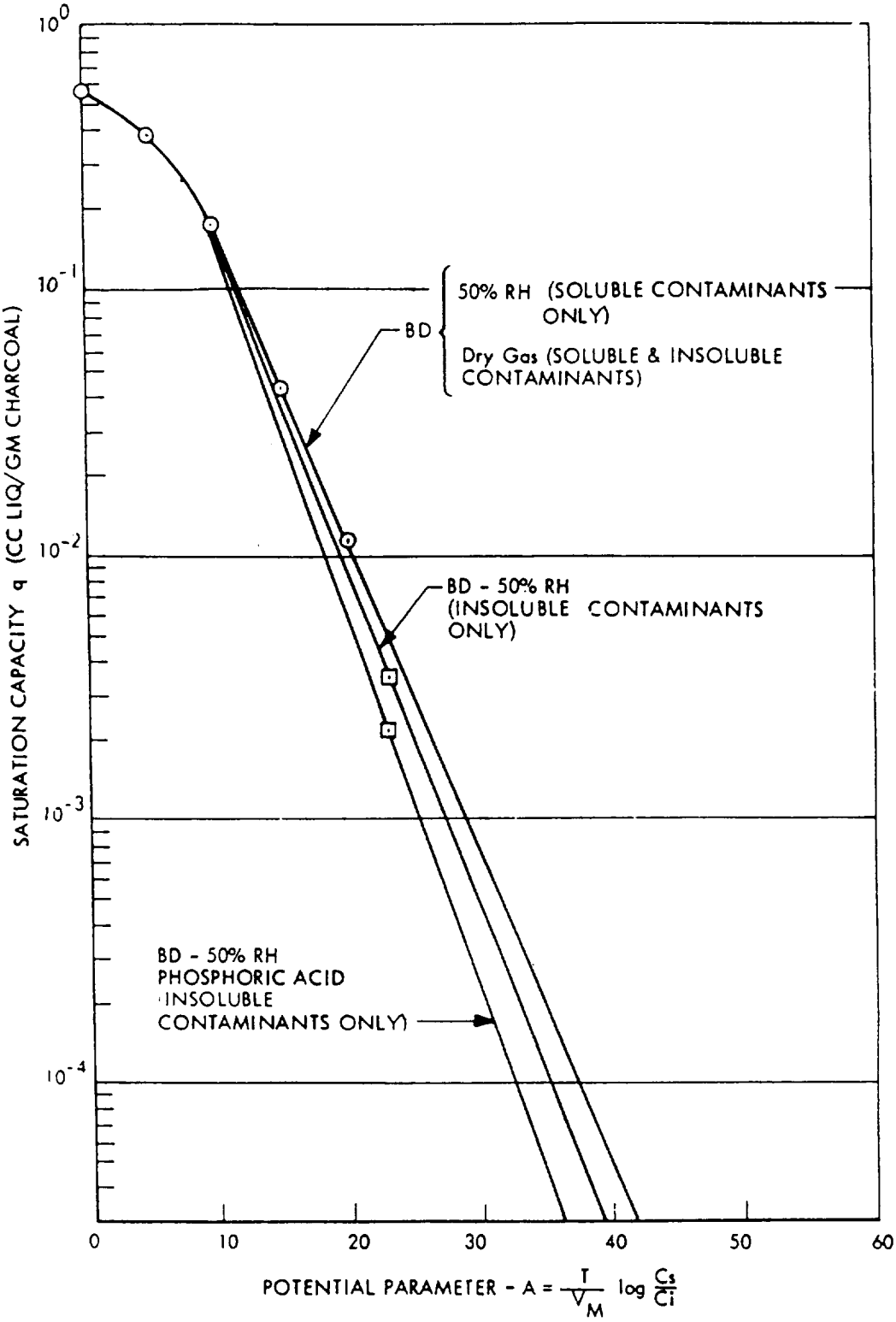


Figure 12. Potential plot for type BD granular activated charcoal.

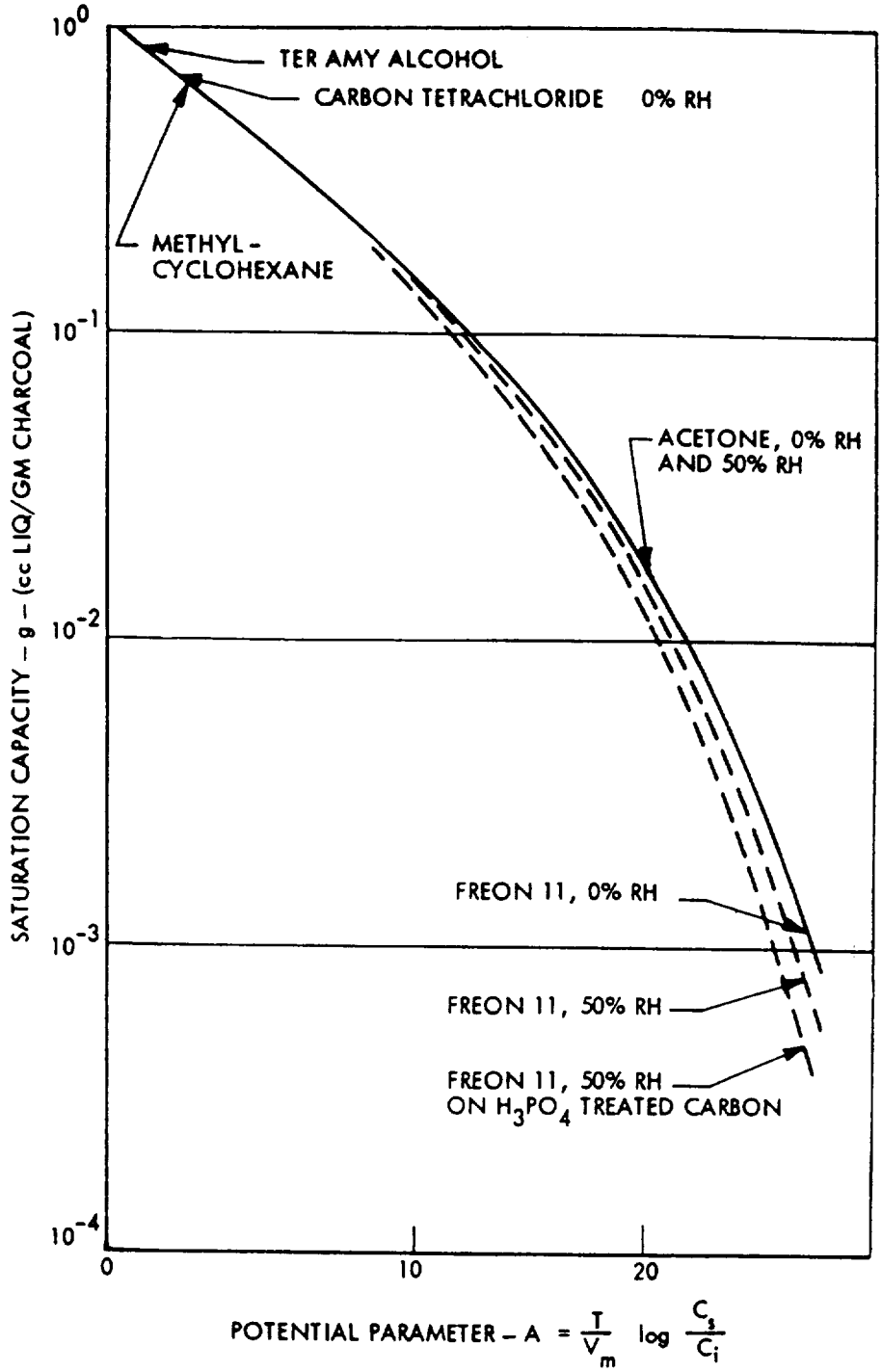


Figure 13. Potential plot showing relative humidity effects.

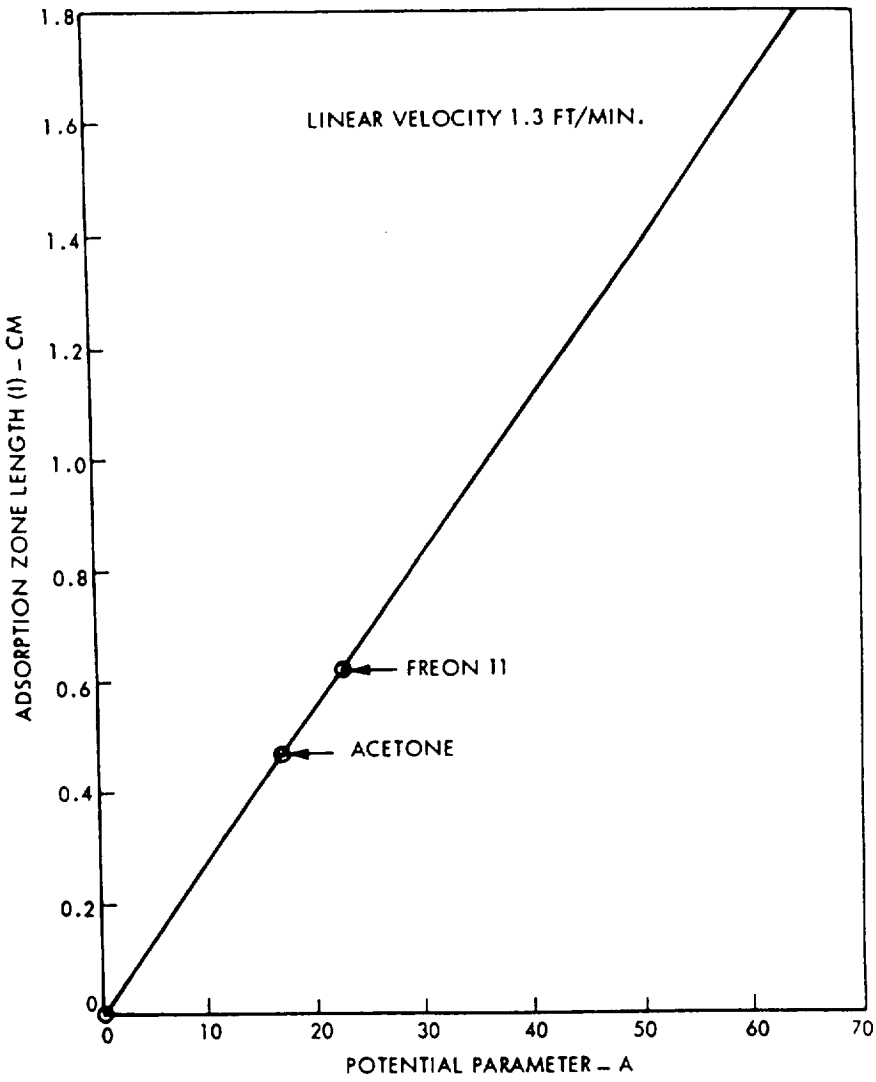


Figure 14. Adsorption zone length as a function of the potential factor.

Adsorption of multiple contaminants by charcoal involves some interaction between the contaminants. This interaction, called blockage or coexistence, reduces the capacity of the charcoal to hold other contaminants. Although the theory for coexistence is complex, experimental data indicates that an additional twenty percent can be added to the saturation zone when the calculation for the adsorption zone is conducted.

Reaction With Specially Treated Charcoals. Some contaminants are not readily removed by granular activated charcoal but can be removed by charcoal which has been specially treated with chemicals that react with the contaminant after adsorption onto the charcoal surface. Two commonly used treated charcoals target ammonia and formaldehyde.

Ammonia Removal. Ammonia is removed by treating granular activated charcoal with phosphoric acid. Usually, phosphoric acid loading is 1.22 mmol/g of charcoal. This results in a requirement of 0.0061 g of charcoal per gram of ammonia to be removed if the reaction goes to completion. The bed removal efficiency is typically 90 to 99 percent per pass for a fresh bed. As the phosphoric acid is

depleted, the efficiency drops, eventually reaching zero. This routine assumes that the removal efficiency is 100 percent if the bed is less than 80 percent utilized. The efficiency for last 20 percent of the bed is calculated using the following sine relationship:

$$\eta_r = \sin\left(m_{\text{charcoal}} - m_{\text{treated charcoal used}}\right)/(0.2)m_{\text{charcoal}}\,, \tag{15}$$

where m_{charcoal} is the mass of charcoal in the bed and $m_{\text{treated charcoal used}}$ is the mass of treated charcoal used.

Formaldehyde Removal. Formaldehyde is removed most efficiently by chromate impregnated charcoal. Manufacturer's data indicates that this charcoal can chemisorb a total amount of formaldehyde equivalent to 5 percent of its weight.[9] Testing at Lockheed Missiles and Space Company, Inc., showed this material's efficiency to drop linearly from 100 to 90 percent for an amount of formaldehyde chemisorbed from 0 to 0.12 percent of the bed weight. Also, if the bed residence time is less than 0.25 s, the removal efficiency drops linearly.

Subroutine RCHBD

The subroutine RCHBD uses the same logic as ACHBD for simulating charcoal adsorption. However, this routine accommodates the geometry of a radial flow charcoal bed.

Subroutine ALIOH

ALIOH simulates removal of acidic contaminants by granular lithium hydroxide and lithium carbonate. The amount of lithium hydroxide consumed per weight of contaminant is calculated from the reaction stoichiometry. This number is input with the contaminant data. Reaction of lithium hydroxide with carbon dioxide to produce lithium carbonate has no noticeable effect on the contaminant removal. The removal efficiency for this device is 100 percent unless the bed is less than 1.905-cm thick or more than 80 percent utilized. The drop in efficiency as the bed is utilized is approximated by the following sine relationship:

$$\eta_r = \sin\left(m_{\text{LiOH}} - m_{\text{LiOH used}}\right)/(0.2)m_{\text{LiOH}}\,, \tag{16}$$

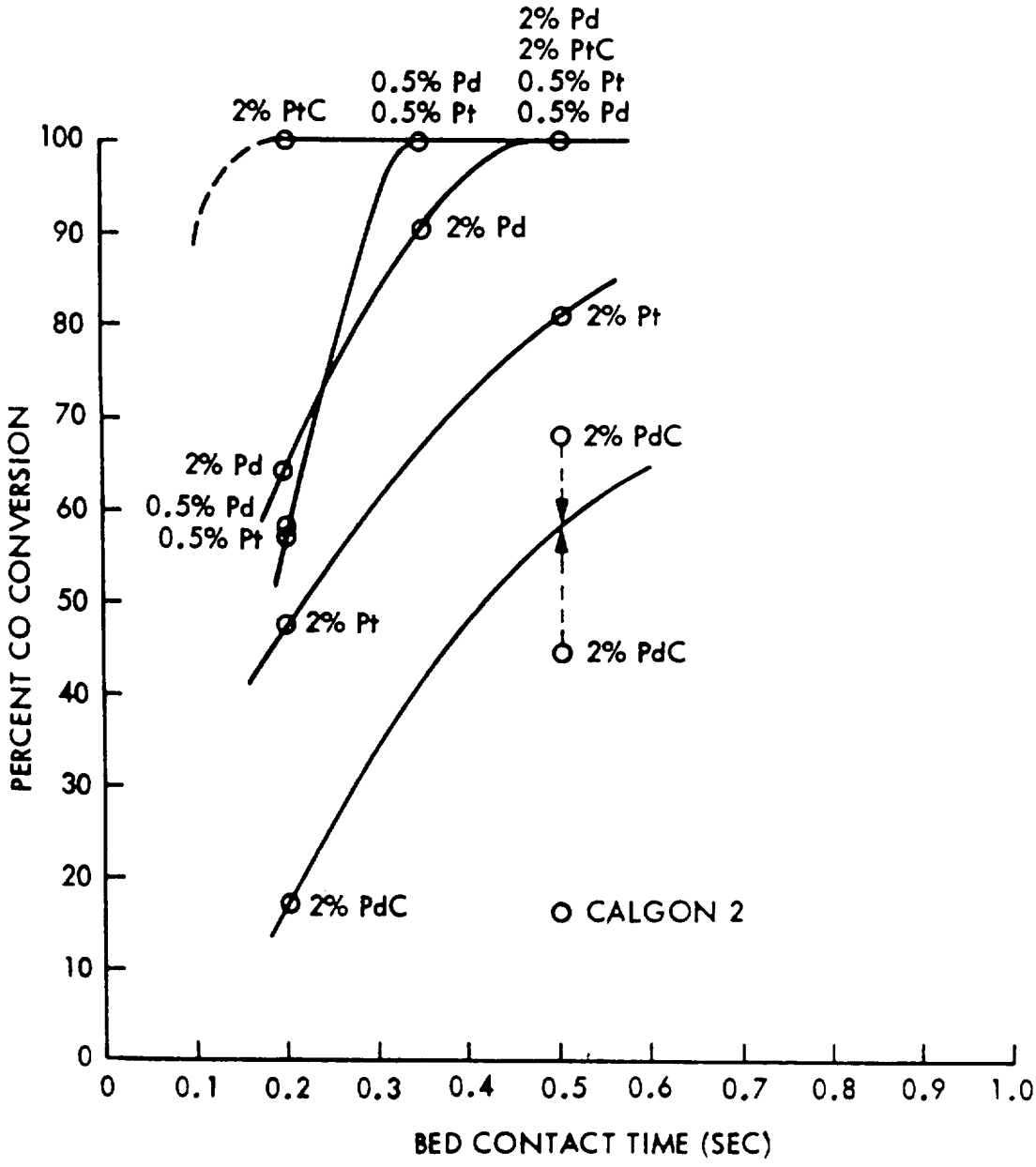
where m_{LiOH} is the mass of the lithium hydroxide bed and $m_{\text{LiOH used}}$ is the mass of lithium hydroxide utilized. Efficiency for a bed less than 1.905-cm thick decreases linearly with thickness.

Subroutine COOXID

Ambient temperature catalytic oxidation of carbon monoxide and hydrogen is simulated by the COOXID subroutine. This routine simulates ambient temperature catalytic oxidation using a granular activated charcoal with 2 weight percent platinum loading. This simulation is effective only for carbon monoxide and hydrogen. Efficiency remains constant at 100 percent per pass unless the residence time falls below 0.2 s. For residence times below 0.2 s, the efficiency decreases linearly according to figure 15.[10]

Subroutine CATBNR

The CATBNR subroutine simulates the destruction of hydrogen, carbon monoxide, methane, and other low molecular weight organic contaminants into carbon dioxide and water vapor using high temperature catalytic oxidation. The degree of oxidation in the oxidizer must be input by the user in the



NOTES:

1. ALL NOBLE METALS ON ALUMINA SUBSTRATE UNLESS OTHERWISE STATED
2. 2% PtC = 2% PLATINUM ON CARBON
3. 2% PdC = 2% PALLADIUM ON CARBON

Figure 15. Noble metal CO catalyst performance.

contaminant data input file. Typically, oxidation efficiency is based on experimental oxidation performance testing. On average, operating the oxidizer at 400 °C (750 °F) provides removal efficiency of 100 percent for most contaminants.

Subroutine CONDHX

The subroutine CONDHX simulates the removal of contaminants by absorption into humidity condensate in a condensing heat exchanger. Some contaminants are removed by this route not only by absorption but also by chemical reaction in the condensate. Ammonia is treated in this manner since it dissociates in water and reacts with dissolved carbon dioxide.[11] All other contaminant removal is simulated using Henry's Law. Using Henry's Law is justified for trace contaminants since their concentrations in the atmosphere approach infinite dilution. Henry's Law correlates the concentration of a contaminant in the atmosphere to its concentration in the liquid phase. The correlation coefficient is the Henry's Law Constant, H, which has units of atmospheres per mole fraction. Equation (17) shows the Henry's Law relationship in which p_c is the contaminant partial pressure in atmospheres, H is the Henry's Law constant in atmospheres per mole fraction, and x is the liquid phase mole fraction.

$$p_c = Hx \,. \tag{17}$$

The simulation assumes that the absorption process is concurrent and that equilibrium is very closely approached. A material balance on this process provides a relationship for the condensate mole fraction shown by:

$$x = y/[(C/A)+(H/P)] \,. \tag{18}$$

In this equation, x is the liquid phase mole fraction, y is the vapor phase mole fraction, C is the condensate mass molar flow rate in mol/h, A is the atmospheric molar flow rate in mol/h, H is the Henry's Law constant in atmospheres, and P is the total pressure in atmospheres. Figure 16 illustrates the absorption process. Based on the cabin concentration, the program calculates the inlet mole fraction based on a 1 atmosphere total pressure. The condensate flow rate and atmosphere flow rate are entered in the device definition data and converted to molar flow rates based on 1 atmosphere pressure and 294 K absolute temperature. The mole fraction of contaminant leaving in the condensate is used to determine the mass of contaminant removed. The removal efficiency is calculated from the ratio of the difference in mass of contaminant entering and mass of contaminant removed to the mass of contaminant entering.

Ammonia removal is treated separately since it reacts chemically with dissolved carbon dioxide in the humidity condensate. According to reference 22, data correlating ammonia partial pressure to liquid phase ammonia concentration for several carbon dioxide atmospheric partial pressures was used to obtain an equation relating liquid and gas phase ammonia composition. This data was obtained by sparging a gas mixture through a volume of water. Figure 17 shows a plot of the result.[12] The carbon dioxide curve corresponding to 666.6 Pa (5 mm Hg) was used to obtain equation (19) which relates ammonia mass per kilogram of condensate to the entering ammonia concentration.

$$m_a = 189.6 C_c^{0.535} \,. \tag{19}$$

In this equation, m_a is the mass of ammonia in milligrams per kilogram of condensate and C_c is heat exchanger inlet ammonia concentration in mg/m^3. This equation is used to determine the ammonia removal efficiency from the mass of ammonia entering and leaving the condensing heat exchanger assembly.

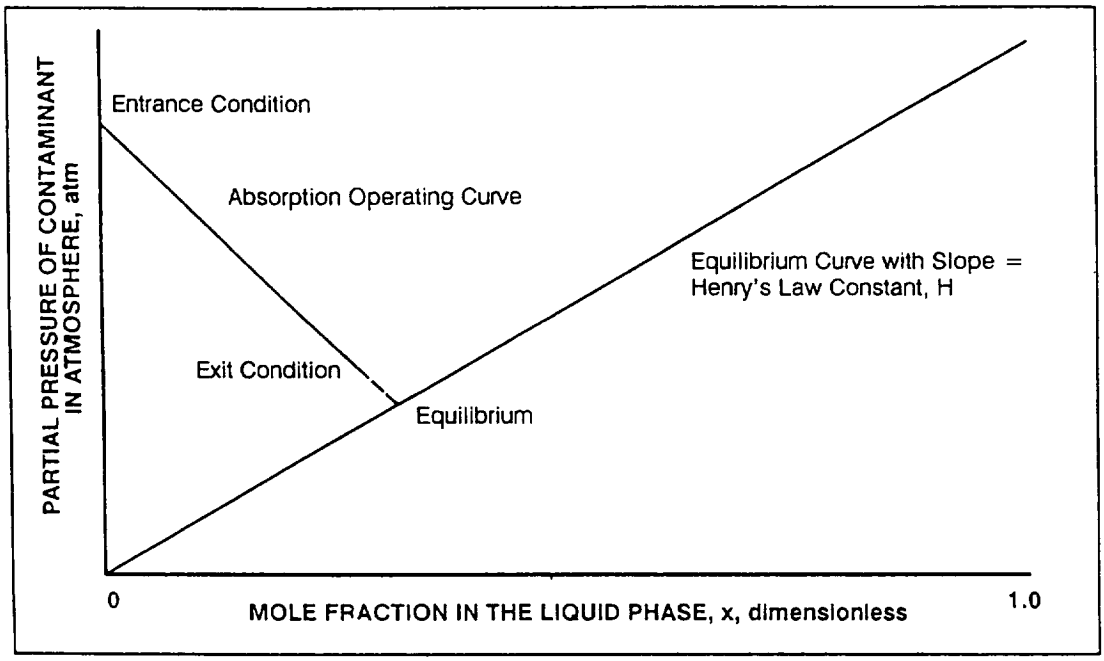


Figure 16. Absorption of contaminants by humidity condensate.

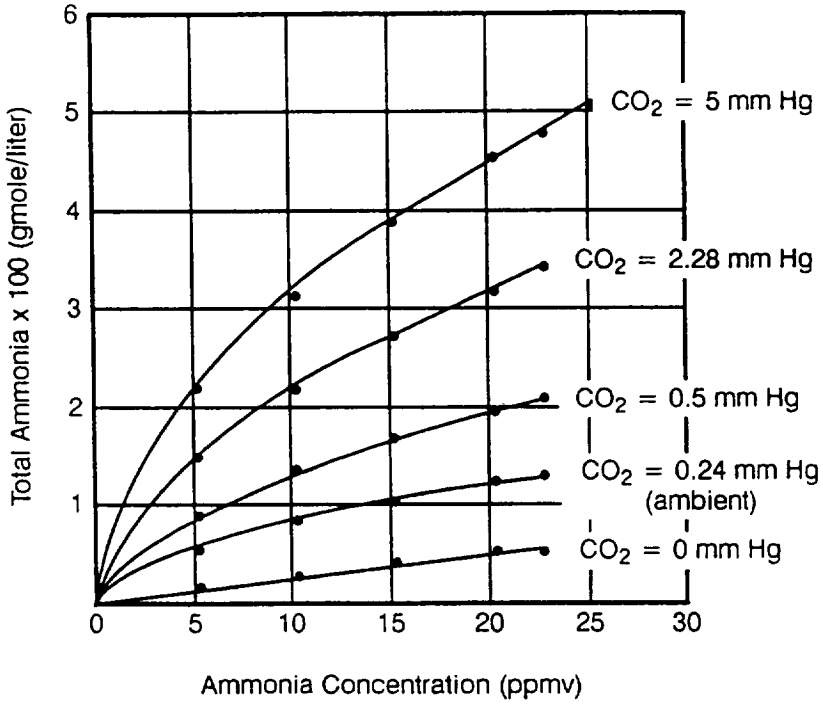


Figure 17. Ammonia solubility in water at varying carbon dioxide concentration.

Data Input and Output Subroutines

The following subroutines regulate the data input and output for each computer simulation run. These subroutines have been designed to allow maximum flexibility for calculated data output to facilitate data analysis reporting.

Subroutine CRIN

Subroutine CRIN is called by MAIN and reads contaminant input data into the contaminant name matrix, NN, and the main calculation matrix, CDI.

Subroutine RRIN

Subroutine RRIN is called by MAIN and reads device definition data and time-dependent data into matrices DD and TT, respectively.

Subroutine CROUT2

CROUT2 is called by MAIN and controls output of the contaminant input data to the printer or computer terminal screen. One row at a time without headings is written to these output devices for the user to review before entering the calculation loop.

Subroutine RROUT2

Subroutine RROUT2 is called by MAIN and controls output of the device definition data and time-dependent data to the printer or computer terminal screen. One row at a time without headings is written to these output devices for the user to review before entering the calculation loop.

Subroutine CROUT

CROUT is called by MAIN and regulates output of matrix CC data during each time increment for diagnostic purposes. This subroutine is called only when print switch No. 5 is set equal to 1.

Subroutine RROUT

Subroutine RROUT is called by MAIN and regulates output of matrix DD for diagnostic purposes. This subroutine is called only when print switch No. 5 is set equal to 1.

Subroutine DATOUT

DATOUT is called by MAIN and serves as the master output regulation routine. Routines contained within PRFANS are called from DATOUT according to the print switch designations made by the user.

Subroutine GROUP

Subroutine GROUP is called by DATOUT and calculates the toxic hazard index according to appendix B. This subroutine also regulates the output for the toxic hazard index for both the standard formatted output and the plot data output.

Subroutine PRFANS

PRFANS contains several subroutines that are called by DATOUT which regulate the output for contaminant concentration data, sum of contaminant masses removed data, and removal device efficiency data. This subroutine regulates output for both the standard formatted output and the plot data output.

Subroutine HEADGS

Subroutines within HEADGS are called by PRFANS subroutines to regulate standard formatted data output headings. Headings are provided for contaminant concentration data, contaminant removal rate data, sum of contaminant masses removed data, and removal device efficiency data.

APPENDIX A

TCCS COMPUTER PROGRAM
VERSION 8.1
FORTRAN CODE LISTING

This appendix contains listings for each major subroutine and the main TCCS computer program. The main program is listed first followed by listings of each subroutine. The subroutines listings are arranged in alphabetical order by name to provide easy reference.

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\FORTRAN\TCC\MAIN.F Options: /C 80 /L /BIJY   03/15/94 13:23:35

   1 C FILE:MAIN.FOR
   2       PROGRAM TCCS81
   3 C     ************************************************************************
   4 C     ************************************************************************
   5 C     *          PROGRAM TO MODEL REMOVAL OF SPACECRAFT                      *
   6 C     *                    GASEOUS CONTAMINANTS                              *
   7 C     *                     VERSION 8.1 Alpha                                *
   8 C     *                       March 15, 1994                                 *
   9 C     ************************************************************************
  10 C     ************************************************************************
  11 C     SUBROUTINES REQUIRED:
  12 C       CAFILL-FILL MATRIX WITH ZEROS
  13 C       RAFILL-FILL MATRIX WITH ZEROS
  14 C       CRIN-READ IN INPUT DATA
  15 C       RRIN-READ IN INPUT DATA
  16 C       CROUT-PRINT OUT INPUT DATA
  17 C       RROUT-PRINT OUT INPUT DATA
  18 C       PCSET-PRECALCULATION SET UP ROUTINE
  19 C       MCALC-MAIN CALCULATION ROUTINE
  20 C       DATOUT-DATA PRINTOUT ROUTINE
  21 C       XXXXX-TIME DEPENDENT DATA ROUTINE
  22 C       REGEN-REGENERATION OF DEVICES ROUTINE
  23 C       SLIOH-SUM LIOH USED IN TIME INCREMENT
  24
  25 C     NOTE:SUBROUTINES USE ADJUSTABLE SIZE ARRAYS
  26 C          WATCH COMPILER OPTIONS/DIMENSIONING IF
  27 C          ANY ARRAY IS LARGER THAN 64K BYTES
  28 C
  29 C     ******************DIMENSION MAIN PROGRAM MATRICES****************
  30 C     NN=CONTAMINANT NAME MATRIX
  31 C     CDI=CONTAMINANT INPUT DATA MATRIX
  32 C     CC=CALCULATON MATRIX
  33 C     DD=DEVICE DEFINITION MATRIX
  34 C     TT=TIME DEPENDENT DATA MATRIX
  35 C
  36 C     PRESENTLY SET TO HANDLE MAXIMUM OF 150 CONTAMINANTS IN MATRICES
  37 C     THIS VALUE =NROW AND IS USED IN ADDRESSING ADJUSTABLE SIZE
  38 C          ARRAYS IN SUBROUTINES
  39 C     NTTROW IS USED FOR MATRIX TT MAXIMUM LENGTH
  40 C
  41       CHARACTER NN(300)*30
  42       REAL CDI(300,23)
  43       REAL CC(300,48)
  44 C     REAL DD(15,21)
  45       REAL DD(15,23)
  46       REAL TT(750,7)
  47
  48
  49 C     NOTE:MUST COMPILE SUBROUTINES PROPERLY FOR ADJUSTABLE SIZE
  50 C     ARRAYS IF A MAIN MATRIX EXCEEDS 65536 BYTES (REALS=4 BYTES)
  51 C
  52 C     ************ DECLARE OTHER TERMS USED IN MAIN PROGRAM  *********
```

```
      CHARACTER FNAME*24,DES*1,FCPLOT*24,FTPLOT*24,FEPLOT*24
      LOGICAL EX
      INTEGER PRTSW1,PRTSW2,PRTSW3,PRTSW4,PRTSW5,PRTSW6,PRTSW7,PRTSW8,
     +  TVAL,IDEVNO,IDEVN1,IDEVN2,IDEVN3,PRTSW9

C     ******* PRINT WELCOME AND PROGRAM VERSION NUMBER ***************
      WRITE (*,9)
  009   FORMAT (1X,'*****************************************************'/
     +          1X,'*              WELCOME TO THE WORLD                 *'/
     +          1X,'*                   OF THE                          *'/
     +          1X,'*  SPACECRAFT ATMOSPHERIC TRACE CONTAMINATION       *'/
     +          1X,'*          CONTROL SIMULATION PROGRAM               *'/
     +          1X,'*              -VERSION 8.1 Alpha-                  *'/
     +          1X,'*                March 15, 1994                     *'/
     +          1X,'*****************************************************')
C
C     ******* DEFINE PROGRAM VARIABLES *********************************
C     LIN=NO. OF LINES OF DATA IN MAT NN & MAT CDI & MAT CC
C     LIN1=NO. LINES OF DATA IN MAT TT
C     LIN2=NO. LINES OF DATA IN MAT DD
C     TN=INCREMENT END TIME (HRS)
C     TN1=INCREMENT BEGINNING TIME (HRS)
C     TMIS=TOTAL MISSION TIME (HRS)
C     NINC=NUMBER OF TIME INCREMENTS ELAPSED
C
C     MAT NN,CC, AND CDI MUST HAVE SAME NO. OF ROWS
C     DIMENSIONS OF MAT DD
      NROW=15
C     NCOL=21
      NCOL=23
C     DIMENSIONS OF MAT CC & ROWS IN MAT NN
      NROW1=300
      NCOL1=48
C     DIMENSIONS OF MAT CDI
      NROW2=NROW1
      NCOL2=23
C     DIMENSIONS OF MAT TT
      NTTROW=750
      NTTCOL=7
C     DEVICE NUMBER FOR OUTPUT DATA (SET TO 6 FOR FORM FEED ON OUTPUT)
      IDEVNO=6
C     DEVICE NUMBER FOR MESSAGE OUTPUT
      IMSGDN=2
C     DEVICE NUMBER FOR CONTAMINANT PLOT DATA
      IDEVN1=10
C     DEVICE NUMBER FOR T-VALUE PLOT DATA
      IDEVN3=11
C     DEVICE NUMBER FOR EFFICIENCY PLOT DATA
      IDEVN2=12
C     ***********   END OF DEFINITION SECTION **************************
C
C     *************** ZERO MATRICES***************************************
C     PUT BLANKS IN NAME MATRIX
  011 CALL CAFILL(NN,1,NROW1)
C     PUT ZEROS IN OTHER MATRICES
      CALL RAFILL(CDI,NROW2,NCOL2)
      CALL RAFILL(CC,NROW1,NCOL1)
      CALL RAFILL(DD,NROW,NCOL)
      CALL RAFILL(TT,NTTROW,NTTCOL)
C
```

```
113 C      ******* READ IN DATA FROM FILES AND PRINT IT IF DESIRED ********
114 C
115    010 WRITE(*,*)'INPUT CONTAMINANT DATA FILE NAME:  '
116        CALL CRIN(NN,CDI,NROW2,NCOL2,LIN)
117    012 WRITE(*,*)'PRINT CONTAMINANT INPUT DATA? (Y/N) '
118        READ(*,'(A)')DES
119        IF((DES.NE.'Y') .AND. (DES.NE.'N')) GOTO 12
120        IF (DES.EQ.'N') GOTO 20
121        CALL CROUT2(NN,CDI,NROW2,NCOL2,1,NCOL2,LIN,1,LIN,IMSGDN)
122 C
123    020 WRITE(*,*)'INPUT DEVICE DEFINITION TABLE FILE NAME:  '
124 C      NOTE: ONLY 16 COLUMNS ARE IN THE INPUT FILE
125        CALL RRIN(DD,NROW,NCOL,16,LIN2)
126    022 WRITE(*,*)'PRINT DEVICE DEFINITION TABLE? (Y/N) '
127        READ(*,'(A)')DES
128        IF((DES.NE.'Y') .AND. (DES.NE.'N'))  GOTO 22
129        IF(DES.EQ.'N') GOTO 30
130        CALL RROUT2(DD,NROW,NCOL,1,16,LIN2,IMSGDN)
131 C
132    030 WRITE(*,*)'INPUT TIME DEPENDENT DATA FILE NAME:  '
133        CALL RRIN(TT,NTTROW,NTTCOL,NTTCOL,LIN1)
134    032 WRITE(*,*)'PRINT TIME DEPENDENT DATA? (Y/N) '
135        READ(*,'(A)')DES
136        IF((DES.NE.'Y') .AND. (DES.NE.'N'))  GOTO 32
137        IF (DES.EQ.'N') GOTO 40
138        CALL RROUT2(TT,NTTROW,NTTCOL,1,NTTCOL,LIN1,IMSGDN)
139 C
140 C      *************READ IN MISSION DATA VARIABLES **********************
141 C      READ IN MISSION TOTAL TIME (HRS)
142    040 WRITE(*,*)'INPUT TOTAL MISSION TIME IN HOURS:  '
143        READ(*,*) TMIS
144 C
145 C      ********** PRINT SWITCH DEFINITION ******************************
146 C     1=RESULTS FOR ONE CONTAMINANT IN PCSET
147 C     2=RESULTS FOR 1 CONT & INCR IN 1/10 INCR CONV ROUTINE (IN MCALC)
148 C     3=CONVERGENCE VALUES IN CONVRG
149 C     4=RESULTS FOR 1 CONT IN MCALC AFTER CAV CALC
150 C     5=MAT CC AND MAT DD AT END OF TIME INCREMENT
151 C     6=PRINT CONC+M.REM+SUM MASS REM+REM EFF(OTHERWISE ONLY CONC DATA)
152 C     7=PRINT OUTPUT WITH NO FORM FEEDS
153 C     8=PRINT ANSWERS DURING EACH ITERATION (IN MAIN PROGRAM) AND CONTROL
PLO
154 C     9=CONTROL PLOT FILE OUTPUT
155 C     TVAL=CONTROL OUTPUT OF GROUP CONTRIBUTION T-VALUE DATA
156          PRTSW1=NINT(DD(2,9))
157          PRTSW2=NINT(DD(2,10))
158          PRTSW3=NINT(DD(2,11))
159          PRTSW4=NINT(DD(2,12))
160          PRTSW5=NINT(DD(2,13))
161          PRTSW6=NINT(DD(2,14))
162          PRTSW7=NINT(DD(2,15))
163          PRTSW8=NINT(DD(2,16))
164 C
165 C      **** MAKE DECISION ON CONCENTRATION AND EFFICIENCY PLOT DATA ****
166        WRITE (*,*) 'DO YOU WISH TO WRITE INCREMENT DATA TO A PLOT FILE?'
167        WRITE (*,*) '    1.  Concentration Data (C)'
168        WRITE (*,*) '    2.  Efficiency Data (E)'
169        WRITE (*,*) '    3.  Both Concentration and Efficiency Data (B)'
170        WRITE (*,*) '    4.  Neither (N)'
171        WRITE (*,*) 'ENTER YOUR SELECTION:
```

```
172         READ (*,'(A)') DES
173        IF (DES.EQ.'C') THEN
174          PRTSW9=1
175        ELSEIF (DES.EQ.'E') THEN
176          PRTSW9=2
177        ELSEIF (DES.EQ.'B') THEN
178          PRTSW9=3
179        ELSE
180          PRTSW9=0
181        ENDIF
182 C
183 C      *************** MAKE DECISION ON T-VALUE OUTPUT *****************
184        WRITE (*,*) 'PRINT GROUP CONTRIBUTION T-VALUE DATA?'
185        WRITE (*,*) '     1.  Print to Normal Output (Y)'
186        WRITE (*,*) '     2.  Print to Normal Output and Plot File (P)'
187        WRITE (*,*) '     3.  Do Not Print (N)'
188        WRITE (*,*) 'ENTER YOUR SELECTION:  '
189          READ (*,'(A)')DES
190        IF (DES.EQ.'Y') THEN
191          TVAL=1
192        ELSEIF (DES.EQ.'P') THEN
193          TVAL=2
194        ELSE
195          TVAL=3
196        ENDIF
197 C
198 C     CHANGE TO NO FORM FEED IF PRTSW7=1
199        IF (PRTSW7.EQ.1) THEN
200          IDEVNO=7
201        ENDIF
202
203 C      ***************  MAKE DECISIONS ON DATA OUTPUT ******************
204 C      THIS IS WHERE ALL PROGRAM OUTPUT DATA FILES ARE OPENED
205 C      THEY MUST BE CLOSED AT THE END OF THE PROGRAM
206
207 C      SECTION WHICH CHECKS FOR EXISTANCE OF OUTPUT FILE & OPENS IT
208    050 WRITE(*,*) ' WRITE OUTPUT TO FILE, PRINTER, SCREEN, OR END?'
209        WRITE (*,*)'      (FILE NAME/LPT1/CON/END) '
210 C      *********** NOTE:  LPT1 OUTPUT REQUIRES 132 COLUMNS ***********
211        READ(*,'(A)') FNAME
212 C      QUIT IF FNAME=END
213        IF(FNAME.EQ.'END') GOTO 999
214        IF((FNAME.NE.'LPT1').AND.(FNAME.NE.'CON')) THEN
215          INQUIRE(FILE=FNAME,EXIST=EX)
216          IF (EX) THEN
217            WRITE(*,*)'FILE EXISTS - OVERWRITE? (Y/N) '
218            READ(*,'(A)') DES
219            IF (DES.NE.'Y') THEN
220              GOTO 50
221            ELSE
222              OPEN(IDEVNO,FILE=FNAME,STATUS='OLD',IOSTAT=IOVAL)
223            ENDIF
224          ELSE
225           OPEN (IDEVNO,FILE=FNAME,STATUS='NEW',IOSTAT=IOVAL)
226          ENDIF
227        ENDIF
228        IF ((FNAME.EQ.'LPT1').OR.(FNAME.EQ.'CON')) THEN
229          OPEN(IDEVNO,FILE=FNAME,IOSTAT=IOVAL)
230        ENDIF
231        IF(IOVAL.NE.0) THEN
```

```
232         CLOSE (IDEVNO)
233         WRITE(*,*)'IOERROR= ',IOVAL
234         GOTO 50
235       ENDIF
236 C
237 C     *** OPEN FILE FOR CONCENTRATION PLOT DATA IF PRTSW9=1 OR 3 ***
238 C
239       IF ((PRTSW9.EQ.1).OR.(PRTSW9.EQ.3)) THEN
240   052   WRITE (*,*) 'FILE NAME FOR CONCENTRATION PLOT DATA OUTPUT? '
241           READ (*,'(A)') FCPLOT
242         INQUIRE (FILE=FCPLOT,EXIST=EX)
243         IF (EX) THEN
244           WRITE (*,*) 'PLOT FILE EXISTS - OVERWRITE? (Y/N) '
245             READ (*,'(A)') DES
246           IF (DES.NE.'Y') THEN
247             GOTO 52
248           ELSE
249             OPEN (UNIT=10,FILE=FCPLOT,STATUS='OLD',IOSTAT=IOVAL)
250           ENDIF
251         ENDIF
252         OPEN (UNIT=10,FILE=FCPLOT,STATUS='NEW',IOSTAT=IOVAL)
253       ENDIF
254       IF (IOVAL.NE.0) THEN
255         CLOSE (UNIT=10)
256         WRITE (*,*) 'IOERROR= ',IOVAL
257         GOTO 52
258       ENDIF
259 C
260 C     ******* OPEN FILE FOR EFFICIENCY PLOT DATA IF PRTSW9=2 OR 3 ******
261 C
262       IF ((PRTSW9.EQ.2).OR.(PRTSW9.EQ.3)) THEN
263   054   WRITE (*,*) 'FILE NAME FOR EFFICIENCY PLOT DATA OUTPUT? '
264           READ (*,'(A)') FEPLOT
265         INQUIRE (FILE=FEPLOT,EXIST=EX)
266         IF (EX) THEN
267           WRITE (*,*) 'PLOT FILE EXISTS - OVERWRITE? (Y/N) '
268             READ (*,'(A)') DES
269           IF (DES.NE.'Y') THEN
270             GOTO 54
271           ELSE
272             OPEN (UNIT=12,FILE=FEPLOT,STATUS='OLD',IOSTAT=IOVAL)
273           ENDIF
274         ENDIF
275         OPEN (UNIT=12,FILE=FEPLOT,STATUS='NEW',IOSTAT=IOVAL)
276       ENDIF
277       IF (IOVAL.NE.0) THEN
278         CLOSE (UNIT=12)
279         WRITE (*,*) 'IOERROR= ',IOVAL
280         GOTO 54
281       ENDIF
282 C
283 C     ********** OPEN FILE FOR T-VALUE PLOT DATA IF TVAL=2 ***********
284 C
285       IF (TVAL.EQ.2) THEN
286   056   WRITE (*,*) 'FILE NAME FOR T-VALUE PLOT DATA OUTPUT? '
287           READ (*,'(A)') FTPLOT
288         INQUIRE (FILE=FTPLOT,EXIST=EX)
289         IF (EX) THEN
290           WRITE (*,*) 'PLOT FILE EXISTS - OVERWRITE? (Y/N) '
```

```
291             READ (*,'(A)') DES
292           IF (DES.NE.'Y') THEN
293             GOTO 56
294           ELSE
295             OPEN (UNIT=11,FILE=FTPLOT,STATUS='OLD',IOSTAT=IOVAL)
296           ENDIF
297         ENDIF
298         OPEN (UNIT=11,FILE=FTPLOT,STATUS='NEW',IOSTAT=IOVAL)
299       ENDIF
300       IF (IOVAL.NE.0) THEN
301         CLOSE (UNIT=11)
302         WRITE (*,*) 'IOERROR= ',IOVAL
303         GOTO 56
304       ENDIF
305 C
306 C
307 C     CALL SYSTEM TIME AND DATE
308 C     THIS MUST BE CALLED ONLY ONCE SO THAT THE TIME AND DATE WILL
309 C       BE THE SAME ON ALL OUTPUT INFORMATION FOR ONE RUN
310       CALL DATTM(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,ISECOND)
311
312 C     SET IPGCTR=COUNTER FOR SEQUENTIAL PAGE NUMBERS ON ALL OUTPUTS
313       IPGCTR=0
314
315 C     ********* CHECK BASIC TIME INCREMENT **************************
316 C     ********* BASIC TIME INCREMENT (HRS)-DD(1,11) *****************
317       BINC=DD(1,11)
318 C     ********* TEST FOR BINC=0 (CAUSES ENDLESS TIME LOOP) **********
319       IF (BINC.EQ.0) THEN
320         CLOSE (IDEVNO)
321         WRITE(*,*)' ERROR-BASIC TIME INCREMENT=0'
322         GOTO 999
323       ENDIF
324
325 C     **************  ZERO INITIAL VARIABLES *************************
326       TN=0
327       TN1=0
328       NINC=0
329 C
330 C     ************************************************************************
331 C     *          PRECALCULATION SET UP ROUTINE                               *
332 C     ************************************************************************
333 C     FOR ALL CONTAMINANTS ONE AT A TIME AT CAV PRED=1E-20, CALC INIT
334 C     DEV EFF AND LOAD IT INTO MAT CC-ALSO CALC CAVPRD(CAV PREDICTED)
335 C   OUTPUTS TO PRECALC SET UP ROUTINE:
336 C    TN1=INCREMENT INITIAL TIME (HRS)
337 C    BINC=BASIC INCREMENT SIZE (HRS) (REF.=DD(1,11)) PASS IN????
338 C    LIN=NO. OF CONT IN MAT CC AND NN
339 C    DD,NROW,NCOL=NAME & SIZE OF MAT DD
340 C    CC,NROW1,NCOL1=NAME & SIZE OF MAT CC
341 C    CDI,NROW2,NCOL2=NAME & SIZE OF MAT CDI
342 C    LIN2=NO. DEVICES IN MAT DD
343 C    NN=NAME OF MAT NN
344 C   INPUTS FROM PRECALC SETUP ROUTINE-SUBROUTINE PCSET:
345 C    PUT IN MAT CC
346 C      CAVPRD=PRED CABIN AV CONC (MG/CU M): =CC(I,2)
347 C      CEQLIB=EQUILIBRIUM CABIN CONT CONC (MG/CU M):=CC(I,3)
348 C      CFINAL=FINAL CABIN CONT CONC (MG/CU M):=CC(I,4)
349 C      PUTS REM EFF FROM DD COL 20 IN CC(I,7-10-13-16 ETC)
350 C      PUTS M.REM IN CC(I,5-8-11-14..)
```

```
351
352       CALL PCSET(TN1,LIN,DD,NROW,NCOL,CC,NROW1,NCOL1,
353      +CDI,NROW2,NCOL2,LIN2,NN,PRTSW1,IMSGDN)
354
355 C     ************* END OF PRECALCULATION SETUP ROUTINE ***************
356
357 C     ***********************************************************************
358 C     *         BEGINNING OF CALCULATION FOR EACH TIME INCREMENT             *
359 C     ***********************************************************************
360   100 CONTINUE
361 C
362 C     *********  INCREASE INCREMENT COUNTER  ***************************
363       NINC=NINC+1
364 C
365 C     *********  SET UP TIME INCREMENT SIZE FOR INCREMENT   ************
366 C
367       IF (NINC.EQ.1) TN=BINC/24
368       IF (NINC.EQ.2) TN=BINC/2
369       IF (NINC.EQ.3) TN=BINC
370       IF (NINC.GT.3) TN=TN+BINC
371 C     CHECK FOR INCREMENT FINAL TIME > MISSION TIME
372       IF(TN.GT.TMIS) TN=TMIS
373 C
374 C     *********  CHECK FOR CHANGES IN BASIC TIME INCREMENT  ************
375       BINCNEW=BINC
376       DO 105 K=1,LIN1
377         IF ((TT(K,1).GE.TN1).AND.(TT(K,1).LT.TN)) THEN
378           IF ((TT(K,4).EQ.1).AND.(TT(K,6).EQ.11)) THEN
379             BINCNEW=TT(K,7)
380           ENDIF
381         ENDIF
382   105 CONTINUE
383       IF (BINC.NE.BINCNEW) THEN
384          TN=TN-BINC+BINCNEW
385          BINC=BINCNEW
386          DD(1,11)=BINC
387       ENDIF
388 C
389 C     STORE PREVIOUS INCREMENT CABIN VOLUME
390       PREVCVOL=DD(1,9)
391 C
392 C     ********* READ TIME DEPENDENT DATA *******************************
393 C     CALL TIME DEPENDENT DATA SUBROUTINE-RINCDD
394       CALL RINCDD(I,TN,TN1,DD,NROW,NCOL,LIN2,
395      +CC,NROW1,NCOL1,CDI,NROW2,NCOL2,LIN,TT,NTTROW,NTTCOL,LIN1)
396 C
397 C     CHECK TO SEE IF CABIN VOLUME HAS CHANGED, AND IF SO
398 C     UPDATE INITIAL CABIN CONCENTRATION FOR NEW VOLUME
399       IF(PREVCVOL.NE.DD(1,9)) THEN
400          DO 200 I=1,LIN
401             CC(I,1)=CC(I,6)/DD(1,9)
402   200    CONTINUE
403       ENDIF
404 C
405 C     ****** LIST INCREMENT NO. AND TIMES TO CONSOLE ******************
406       IF((FNAME.NE.'CON').OR.(PRTSW8.NE.1))THEN
407         OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
408          WRITE(IMSGDN,65)NINC,TN1,TN
409   065     FORMAT (1X,'INCR NO.= ',I5,' BEGIN & END TIMES (hours)=',
410      +    F9.3,2X,F9.3)
```

```
411         CLOSE (IMSGDN)
412       ENDIF
413
414 C     ****STORE ORIGINAL Q DEVICE IN DD COL 7 (TAKEN FROM DD COL 2)****
415       DO 110 J=1,LIN2
416         DD(J,7)=DD(J,2)
417   110 CONTINUE
418
419 C     ***********  CHECK FOR REGENERATION IN TIME INCREMENT ***********
420 C     CALL REGENERATION SUBROUTINE REGEN
421       CALL REGEN(TN,TN1,DD,NROW,NCOL,CC,NROW1,NCOL1,
422      +CDI,NROW2,NCOL2,LIN,LIN2,IMSGDN)
423
424 C     ********** CALL MAIN CALCULATION SUBROUTINE  *********************
425 C    OUTPUTS TO MAIN CALC SUBROUTINE-MCALC:
426 C     I=CONTAMINANT NO.
427 C     TN,TN1 =INCREMENT END & BEGINNING TIME (HRS)
428 C     DD,NROW,NCOL=NAME & DIM OF MAT DD
429 C     CC,NROW1,NCOL1=NAME & DIM OF MAT CC
430 C     CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
431 C     NN=NAME OF MAT NN
432 C     LIN=NUMBER OF CONTAMINANTS IN MAT NN & CDI
433 C     LIN2=NO. DEVICES IN MAT DD
434 C    INPUTS FROM MAIN CALC ROUTINE-MCALC:
435 C    TO MAT CC
436 C     PUTS CAVCLC,CEQLIV,&CFINAL IN CC(I,2-3 &4)
437 C     PUTS REM EFF FROM DD COL20 IN CC(I,7-10-13 ETC)
438 C     PUTS M.REM FOR EACH DEV FROM DD COL21 IN CC(I,6-9-12 ETC)
439 C     PUTS SUM MASS REM FOR EACH DEV IN CC(I,8-11-14 ETC)
440
441       CALL MCALC(I,TN,TN1,DD,NROW,NCOL,
442      +CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,LIN,LIN2,
443      +PRTSW2,PRTSW3,PRTSW4,IMSGDN)
444
445 C     ************ CALCULATE LIOH USED IN INCREMENT ********************
446 C     CALL LIOH REMOVAL SUBROUTINE SLIOH
447       CALL SLIOH(TN,TN1,DD,NROW,NCOL,CC,NROW1,NCOL1,
448      +CDI,NROW2,NCOL2,LIN,LIN2)
449 C
450 C     ************ RESTORE DEVICE FLOW *********************************
451 C     RESTORE ORIGINAL DEVICE FLOW RATE FROM DD COL 7 TO DD COL 2
452       DO 120 J=1,LIN2
453         DD(J,2)=DD(J,7)
454   120 CONTINUE
455
456 C     ********* PRINTOUT OF DATA FOR EACH TIME INCREMENT **************
457 C     IF PRTSW5=1 THEN PRINT MAT DD+MAT CC INFO FOR THIS CONTAMINANT
458       IF (PRTSW5.EQ.1)  THEN
459 C       OPEN (IMSGDN,FILE='CON',IOSTAT=IOVAL)
460 C         WRITE(IMSGDN,*)'PRINTOUT FOR MAT CC & DD AT END OF TIME INCR'
461 C         WRITE(IMSGDN,*)'INFO FROM MAT CC'
462 C       CLOSE(IMSGDN)
463 C       CALL CROUT(NN,CC,NROW1,NCOL1,1,NCOL1,LIN,1,LIN,IMSGDN)
464         CALL CROUT(NN,CC,NROW1,NCOL1,1,NCOL1,LIN,1,LIN,IMSGDN,NINC,
465      +   FNAME,IDEVNO,IOVAL)
466 C       OPEN (IMSGDN,FILE='CON',IOSTAT=IOVAL)
467 C         WRITE(IMSGDN,*)'INFO FROM MAT DD'
468 C       CLOSE(IMSGDN)
469 C       CALL RROUT(DD,NROW,NCOL,1,NCOL,LIN2,IMSGDN)
470         CALL RROUT(DD,NROW,NCOL,1,NCOL,LIN2,IMSGDN,FNAME,IDEVNO,IOVAL)
```

```
      ENDIF
C     ********** REGULAR PRINTOUT OF DATA FOR EACH INCREMENT ********
      IF (PRTSW8.EQ.1) THEN
        CALL DATOUT(TN,TN1,LIN,DD,NROW,NCOL,CC,NROW1,NCOL1,
     +  CDI,NROW2,NCOL2,LIN2,NN,PRTSW6,PRTSW8,PRTSW9,
     +  IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR,
     +  TVAL,FCPLOT,IDEVN1,IDEVN3,IDEVN2)

        IF (IOVAL.NE.0) THEN
          CLOSE (IDEVNO)
          WRITE(*,*)'PROGRAM DATA OUTPUT ERROR IN INCREMENT = ',NINC
          GOTO 999
        ENDIF
      ENDIF
C     ********* CONTROLS PLOT DATA OUTPUT IF PRTSW8=0 ****************
      IF (PRTSW8.EQ.0) THEN
        IF ((PRTSW9.GT.0).OR.(TVAL.EQ.2)) THEN
          CALL DATOUT (TN,TN1,LIN,DD,NROW,NCOL,CC,NROW1,NCOL1,
     +    CDI,NROW2,NCOL2,LIN2,NN,PRTSW6,PRTSW8,PRTSW9,
     +    IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,
     +    IPGCTR,TVAL,FCPLOT,IDEVN1,IDEVN3,IDEVN2)
        ENDIF
      ENDIF
C
C     ********************** CHECK FOR END OF MISSION *****************
      IF(TN.GE.TMIS) THEN
C       END TIME LOOP
        CONTINUE
      ELSE
C     ********* UPDATE FOR NEXT TIME INCREMENT AND REPEAT ************
C       SET TFINAL FOR THIS INCR = TINIT FOR NEXT INCR
          TN1=TN
C       SET CFINAL FOR INCR=CINIT FOR NEXT INCR-ALL CONTAMINANTS
          DO 130 I=1,LIN
            CC(I,1)=CC(I,4)
  130     CONTINUE
        GOTO 100
      ENDIF
C
C     ******************************************************************
C     *          END OF CALCULATION FOR EACH TIME INTERVAL             *
C     ******************************************************************

C     ****** PRINT FINAL ANSWERS AT END OF MISSION IF DESIRED *********
  900 CONTINUE
      IF (PRTSW8.EQ.0) THEN
      CALL DATOUT(TN,TN1,LIN,DD,NROW,NCOL,CC,NROW1,NCOL1,
     +    CDI,NROW2,NCOL2,LIN2,NN,PRTSW6,PRTSW8,PRTSW9,
     +    IDEVNO,-1,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR,
     +    TVAL,FCPLOT,IDEVN1,IDEVN3,IDEVN2)

        IF (IOVAL.NE.0) THEN
          CLOSE (IDEVNO)
          WRITE(*,*)'PROGRAM DATA OUTPUT ERROR - FINAL PRINTOUT'
          GOTO 999
        ENDIF
      ENDIF
C     ************ CLOSE ALL PROGRAM OUTPUT FILES ******************
      CLOSE (IDEVNO)
```

```
  531         CLOSE (IDEVN1)
  532         CLOSE (IDEVN3)
  533         CLOSE (IDEVN2)
  534 C
  535    999  CONTINUE
  536         WRITE(*,*)'DO YOU WISH TO RUN ANOTHER CASE? (Y/N)'
  537           READ (*,'(A)')DES
  538         IF (DES.EQ.'Y') THEN
  539           GO TO 011
  540         ENDIF
  541 C       ************************************************************************
  542 C       *                          END OF MAIN PROGRAM                         *
  543 C       ************************************************************************
  544         END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\FORTRAN\TCC\ACHBD. Options: /C 80 /L /BIJY   03/15/94 10:38:54

C     ******************************************************
C     * FILE:ACHBD.FOR                                     *
C     * SUBROUTINE FOR REM EFF-AXIAL FLOW CHARCOAL BED     *
C     * DOESNT ALLOW FOR DESORPTION                        *
C     ******************************************************
C
      SUBROUTINE ACHBD (TN,TN1,CIIN,TCABIN,COEXIS,BEDQ,EMAX,BEDL,
     +BEDDIA,DENCH,TRTTYP,DCONT,VMOL,MW,VCONC,SOL,SMR,EFF,RH)
C     OUTPUT:
C       EFF=BED REMOVAL EFF(DEC)
C     INPUTS:
C       TN,TNI=INCREMENT INITIAL AND FINAL TIMES(HR)
C       CIIN=BED INLET CONT CONC (MG/CU M)
C       TCABIN=CABIN TEMP (DEG K)
C       COEXIS=COEXISTANCE FACTOR
C       BEDQ=BED FLOW RATE(CU M/HR)
C       EMAX=MAXIMUM BED EFF (DEC)
C       BEDL=BED LENGTH (M)
C       BEDDIA=BED DIAMETER (M)
C       DENCH=DENSITY OF CHARCOAL IN BED (KG/CU M)
C       TRTTYP=BED TREATMENT TYPE(1=CI CHAR,2=PHOS ACID, OTHER
C       #=NONE)
C       DCONT=CONT LIQUID DENSITY (GM/CC)
C       VMOL=CONT MOLAR VOL(GM/CC)
C       MW=CONT MOLECULAR WGT
C       VCONC=CONT VAPOR CONCENTRATION AT TCABIN (MG/CU M)
C       SOL=HENRY'S LAW CONSTANT FOR WATER SOLUBILITY
C       (ATM/MOL FRACTION)
C       SMR=SUM OF CONT MASS STORED IN BED(MG)-FROM LAST INCR
C
      REAL LPREV, LAVN1, LUTIL, LIMM, LAVAV, LADS,MW
      INTEGER FACID,FCI
C
C     SET CIN=CIIN (THIS PREVENTS CIN FROM BEING PASSED BACK UP
C              TO OTHER SUBROUTINES IF IT IS SET TO 1E-20)
      CIN=CIIN

C     BED TREATMENT LOGIC
C       FACID=FLAG IF BED IS TREATED WITH PHOSPHORIC ACID (Y=1
C       N=0)
C       FCI=FLAG FOR CI CHAR IN BED (REMOVES FORMALDAHYDE)
      IF (NINT(TRTTYP).EQ.2) THEN
        FACID=1
        FCI=0
      ELSEIF (NINT(TRTTYP).EQ.1) THEN
        FACID=0
        FCI=1
      ELSE
        FACID=0
        FCI=0
      ENDIF
C
C     TEST FOR NO BED FLOW (BEDQ=<0) OR
C     TN-TN1<=0;BEDL,BEDDIA,DENCH=0
      IF((BEDQ.LE.0).OR.(TN-TN1.LE.0).OR.(BEDL.LE.0).OR.(BEDDIA.LE.0)
     +.OR.(DENCH.LE.0)) THEN
        EFF=0
```

```
 58          GOTO 199
 59        ENDIF
 60 C      TEST FOR CI CHARCOAL AND FORMALDEHYDE(FCI=1 AND MW=30.03
 61        IF((MW.EQ.30.03).AND.(FCI.EQ.1)) THEN
 62          CALL CICH(EFF,EMAX,BEDL,BEDDIA,DENCH,SMR,BEDQ)
 63          GOTO 199
 64        ENDIF
 65 C
 66 C      TEST FOR AMMONIA AND H3PO4 ACID ON CHAR (FACID=1 AND
 67 C      MW=17.0)
 68        IF ((MW.EQ.17.0).AND.(FACID.EQ.1)) THEN
 69          CALL ACIDCH(EFF,EMAX,BEDL,BEDDIA,DENCH,SMR)
 70          GOTO 199
 71        ENDIF
 72 C
 73 C      TEST FOR MOL VOL=0 (NO CHAR REMOVAL)
 74        IF (VMOL.EQ.0) THEN
 75          EFF=0
 76          GOTO 199
 77        ENDIF
 78 C
 79 C      CHARCOAL REMOVAL EFFICIENCY CALCULATION
 80 C      SUPERFICIAL BED VEL(FT/MIN)
 81        BEDVEL=BEDQ*.06960/BEDDIA**2
 82 C      TEST FOR CIN TOO SMALL IN AVAL CALC
 83        IF (CIN.LT.1E-20) CIN=1E-20
 84        AVAL=(TCABIN/VMOL)*LOG10(VCONC/CIN)
 85 C      ADS ZONE LENGTH FOR 90% REMOVAL (M)
 86        LADS=AVAL*.000275*(BEDVEL/1.3)**.8
 87 C      GET QI(CC LIQ CONT/GM CHAR)
 88        CALL FQI(AVAL,QI,FACID,SOL,RH)
 89 C      LENGTH OF BED PREVIOUSLY USED BY CONT AT THIS C INLET (M)
 90        LPREV=SMR*1.273E-6*COEXIS/(DCONT*DENCH*BEDDIA**2*QI)
 91 C      RATE OF BED USAGE (M BED/ MG CONT)
 92        LIMM=1.273E-6*COEXIS/(DCONT*DENCH*BEDDIA**2*QI)
 93 C      LENGTH OF BED AVAILABLE FOR ADS ZONE AT BEGINNING OF
 94 C      INCREMENT (M)
 95        LAVN1=BEDL-LPREV
 96        IF (LAVN1.LT.0) LAVN1=0
 97 C      FIX HERE IF DESORPTION IS DESIRED
 98        IF (LAVN1/LADS.GT.20) THEN
 99          EFFAV=EMAX
100        ELSE
101 C        INIT INCR EFF BASED ON C IN AND BED L AVAIL AT BEG OF
102 C      INCR (DEC)
103          EFAVN1=EMAX*(1-EXP(-2.3025851*LAVN1/LADS))
104 C        LOOP FOR EFFICIENCY
105          EFFAV=EFAVN1
106          DO  399 J=1,10,1
107 C          LENGTH OF BED UTILIZ IN INCR (M)
108            LUTIL=CIN*BEDQ*EFFAV*(TN-TN1)*LIMM
109            IF (LUTIL.GT.LAVN1) THEN
110              GOTO 299
111            ELSE
112 C            AVERAGE BED LENGTH AVAIL (M)
113              LAVAV=LAVN1-LUTIL/2
114              IF ((LAVAV/LADS).GE.20) THEN
115                EFFAV=EMAX
116                GOTO 299
117              ELSE
```

```
 118 C              AV EFF BASED ON AV BED L AVAIL (DEC)
 119                EFFAV=EMAX*(1-EXP(-2.3025851*LAVAV/LADS))
 120              ENDIF
 121            ENDIF
 122   399    CONTINUE
 123   299  ENDIF
 124 C      MAX EFF BASED ON C IN AND RATE OF BED USAGE (DEC)
 125        EFFMAX=LAVN1/(CIN*BEDQ* (TN-TN1)*LIMM)
 126        IF (EFFAV.GT.EFFMAX) EFFAV=EFFMAX
 127        IF (EFFAV.LT.0) EFFAV=0
 128        IF (EFFAV.GT.EMAX) EFFAV=EMAX
 129 C      EFF=ACTUAL EFF OUTPUT FROM SUBROUTINE
 130        EFF=EFFAV
 131 C      REMOVE THIS CHECK IF DESORPTION IS ADDED
 132   199  IF (EFF.LT.0)  EFF=0
 133        IF (EFF.GT.EMAX) EFF=EMAX
 134        RETURN
 135        END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0

 136 C      ******************************************************
 137 C      * SUBROUTINE ACIDCH   -  CALCULATES REMOVAL EFF      *
 138 C      * BED WITH NH3 AND 1.22 MILLIMOLE H3PO4 ON CHAR      *
 139 C      ******************************************************
 140        SUBROUTINE ACIDCH(EFF,EMAX,BEDL,BEDDIA,DENCH,SMR)
 141 C      OUTPUTS
 142 C        EFF=OUTPUT REMOVAL EFF (DEC)
 143 C      INPUTS
 144 C        EMAX=MAXIMUM BED REMOVAL EFF (DEC)
 145 C        BEDL=BED LENGTH (M)
 146 C        BEDDIA=BED DIAMETER (M)
 147 C        DENCH=CHARCOAL DENSITY(KG/CU M)
 148 C        SMR=SUM OF MASS OF CONT REMOVED AT BEG OF INCR (MG)
 149 C
 150 C      FOR AMMONIA CAPACITY AT SMAC
 151 C      CHAR USED (KG)
 152          CHRUSD=1.6E-4*SMR
 153 C      CHAR BED WGT(KG)
 154        BEDWGT=BEDL*BEDDIA**2*.785*DENCH
 155        IF (CHRUSD.LT.0.8*BEDWGT) THEN
 156          EFF=EMAX
 157        ELSE
 158          EFF=EMAX*SIN((BEDWGT-CHRUSD)*1.57/(BEDWGT*0.2))
 159        ENDIF
 160 C      PREVENTS NEGATIVE EFF FOR REACTION
 161 C      IF (EFF.LT.0) EFF=0
 162        IF(EFF.GT.EMAX) EFF=EMAX
 163        RETURN
 164        END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0

 165 C      ******************************************************
 166 C      * SUBROUTINE CICH    -  CALCULATES REMOVAL EFF       *
 167 C      *   FOR FORMALDELYDE AND CI CHAR BED                 *
 168 C      ******************************************************
 169        SUBROUTINE CICH(EFF,EMAX,BEDL,BEDDIA,DENCH,SMR,BEDQ)
```

```
C      OUTPUTS
C        EFF=OUTPUT REMOVAL EFF (DEC)
C      INPUTS
C        EMAX=MAXIMUM BED REMOVAL EFF (DEC)
C        BEDL=BED LENGTH (M)
C        BEDDIA=BED DIAMETER (M)
C        DENCH=CHARCOAL DENSITY(KG/CU M)
C        SMR=SUM OF MASS OF CONT REMOVED AT BEG OF INCR (MG)
C        BEDQ=BED FLOW RATE (CU M/HR)
C
       BEDWGT=BEDL*BEDDIA**2*.785*DENCH
C      PERCENT OF BED WEIGHT CONSUMED (DEC)
       PBWGT=SMR/(BEDWGT*1E6)
       IF(PBWGT.LT..0012) THEN
         EFF=1-PBWGT*83.3
       ELSE
         EFF=.9*COS(PBWGT*1.57/.05)
       ENDIF
C      BED RESIDENCE TIME (SEC)
       BREST=BEDL*BEDDIA**2*3600/(BEDQ*1.273)
       IF(BREST.LT.0.25)THEN
         EFF=EFF*BREST/.25
       ENDIF                    -
C      PREVENTS NEGATIVE EFF FOR REACTION
C      IF (EFF.LT.0) EFF=0
       IF(EFF.GT.EMAX) EFF=EMAX
       RETURN
       END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0

C      ******************************************************
C      * SUBROUTINE FQI   -   FINDS QI, THE ACID TREATED    *
C      * CHARCOAL CAPACITY FOR A CONTAMINANT AT 0 TO 75% RH *
C      * (CC LIQ CONT/GM CHAR)                              *
C      ******************************************************
       SUBROUTINE FQI(AVAL,QI,FACID,SOL,RH)
       INTEGER FACID
C      OUTPUTS
C        QI=CHARCOAL CAPACITY (CC LIQ CONT/GM CHAR)
C      INPUTS
C        AVAL= A VALUE OF CONTAMINANT
C        FACID= FLAG FOR ACID TREATED CHAR IN BED (Y=1 N=0)
C        SOL=CONTAMINANT HENRY'S LAW CONSTANT (ATM/MOL FRACTION)
C        RH=RELATIVE HUMIDITY (%)
C
       IF (RH.LT.0) RH=0
C      CARBON CAPACITY DATA NOT AVAILABLE ABOVE 75% RH
       IF (RH.GT.75) RH=75
C
       IF (AVAL.LT.0)  AVAL=0
C      A VALUE .GT. 8 AND .LT. 200
       IF ((AVAL .GT. 8) .AND. (AVAL.LT.200)) THEN
C      SOLUBLE CONTAMINANTS
            IF(SOL.GT.0.AND.SOL.LT.5) THEN
               QI=2.1*EXP(-0.31*AVAL)
C      INSOLUBLE CONTAMINANTS (IF HENRY'S LAW CONSTANT FOR A CONTAM.
C      IS NOT AVAILABLE A 0 VALUE IS ASSIGNED AND CONTAMINANT IS
C      CONSIDERED WATER INSOLUBLE)
```

```
 226             ELSE
 227                IF(RH.LE.50) THEN
 228                   QI=(0.000096*RH**2-0.0188*RH+2.11)*EXP(-0.31*AVAL)
 229                ELSEIF (RH.GT.50) THEN
 230                   QI=(0.000096*RH**2-0.0188*RH+2.11)*
 231      +               EXP(-AVAL*(0.25+0.0012*RH))
 232                END IF
 233             ENDIF
 234 C      A VALUE .LE. 8
 235        ELSEIF (AVAL.LE.8) THEN
 236           IF(SOL.GT.0.AND.SOL.LT.5) THEN
 237              QI=0.5-AVAL*0.0405
 238           ELSE
 239              QI=-0.0000128*RH**2-0.00264*RH+0.5+(0.00000112*RH**2+
 240      +          0.000208*RH-0.0405)*AVAL
 241           ENDIF
 242 C      AVAL .GE. 200
 243        ELSE
 244          QI=1E-20
 245        ENDIF
 246        RETURN
 247        END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\ALIOH.F Options: /C 80 /L /BY     05/21/92 12:54:48
    1 C      ****************************************************************
    2 C      *                SUBROUTINE  ALIOH - AXIAL FLOW BED            *
    3 C      *           CALCULATES REMOVAL EFFICIENCY FOR LIOH             *
    4 C      ****************************************************************
    5 C
    6          SUBROUTINE ALIOH(TN,TN1,EMAX,BEDL,DENLI,BEDDIA,RWUTLI,SWUTLI,
    7         +REMFCT,EFF)
    8 C
    9 C   INPUTS:
   10 C     TN=FINAL INCREMENT TIME (HRS)
   11 C     TN1=INITIAL INCREMENT TIME (HRS)
   12 C     EMAX=MAXIMUM POSSIBLE REMOVAL EFFICIENCY (DEC)
   13 C     BEDL=BED LENGTH (M)
   14 C     DENLI=LIOH DENSITY (KG/CU M)
   15 C     BEDDIA=BED DIAMETER (M)
   16 C     RWUTLI=RATE OF LIOH USAGE FOR ALL CONTAMINANTS FROM LAST INCR(KG/HR)
   17 C     SWUTLI=SUM OF WEIGHT OF LIOH UTILIZED FROM LAST INCR(KG)
   18 C     REMFCT=LIOH REMOVAL FACTOR (LB LIOH/LB CONTAMINANT)
   19 C   OUTPUTS:
   20 C     EFF=REMOVAL EFFICIENCY (DEC)
   21 C
   22 C   IF CONT DOESNT REACT WITH LIOH OR BEDL<=0 OR BED DIA <=0 OR
   23 C     DENLI <=0 THEN REM EFF =0
   24           IF((REMFCT.LE.0).OR.(BEDL.LE.0).OR.(BEDDIA.LE.0).OR.(DENLI.LE.0))
   25          + THEN
   26             EFF=0
   27           ELSE
   28 C           BED WEIGHT (KG)
   29             BEDWGT=BEDL*(BEDDIA)**2*.785*DENLI
   30 C           TOTAL WEIGHT OF LIOH UTILIZED AT AVERAGE TIME IN INCREMENT (KG)
   31             TWUTLI=SWUTLI+RWUTLI*(TN-TN1)/2
   32             IF (TWUTLI/BEDWGT.LE.0.8) THEN
   33               EFF=EMAX
   34             ELSE
   35               EFF=EMAX*SIN((BEDWGT-TWUTLI)*1.57/(BEDWGT*0.2))
   36             ENDIF
   37             IF (BEDL.LT.0.0191) THEN
   38               EFF=EFF*BEDL/0.0191
   39             ENDIF
   40           ENDIF
   41           IF(EFF.LT.0) EFF=0
   42           IF(EFF.GT.EMAX) EFF=EMAX
   43           RETURN
   44           END
   45 C         ***********  END OF SUBROUTINE ALIOH ****************************

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\CAFILL. Options: /C 80 /L /BY     05/21/92 12:54:53
   1 C      ********************************************************************
   2 C      *         SUBROUTINE CAFILL                                        *
   3 C      *  SUBROUTINE TO FILL ADJUSTABLE SIZE CHAR ARRAY WITH BLANKS       *
   4 C      ********************************************************************
   5        SUBROUTINE CAFILL(NN,NROW,NCOL)
   6        INTEGER NROW,NCOL
   7        CHARACTER NN(NROW,NCOL)*30
   8
   9 C   NN=ARRAY NAME-ARRAY HAS 30 CHARACTERS
  10 C   NROW=NUMBER OF ROWS IN ARRAY (INTEGER)
  11 C   NCOL=NUMBER OF COLUMNS IN ARRAY(INTEGER) 12
  13        DO 110 I=1,NROW
  14        DO 100 J=1,NCOL
  15         NN(I,J)=' '
  16  100   CONTINUE
  17  110   CONTINUE
  18        RETURN
  19 C      ************* END OF SUBROUTINE CAFILL  ***************************
  20        END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\CALCM.F Options: /C 80 /L /BY     05/21/92 12:54:56
   1 C      ****************************************************************
   2 C      *                  SUBROUTINE    CALCM                         *
   3 C      * SUBROUTINE TO CALCULATE SUM OF MASSES REMOVED BY ALL DEVICES *
   4 C      * USES CAV CABIN, REM EFF DD(J,20), & M.GEN DD(J,19) TO CALC   *
   5 C      * DEVICE CIN & COUT DD(J,17)& DD(J,18),M.REM CABIN+DEV DD(J,21)*
   6 C      ****************************************************************
   7
   8        SUBROUTINE CALCM(DD,NROW,NCOL,CAV,SMGEN,SMREM,TN,LIN2)
   9        INTEGER NROW,NCOL,LIN2
  10        REAL DD(NROW,NCOL)
  11 C
  12 C SUBROUTINES REQUIRED:
  13 C  NONE
  14
  15 C INPUTS:
  16 C  CAV=CABIN CONT AVERAGE CONCENTRATION (MG/CU M)
  17 C  DD,NROW,NCOL=NAME AND SIZE OF MAT DD
  18 C  TN=INCREMENT FINAL TIME (HRS)
  19 C  LIN2=NO. OF DEVICES IN MAT DD
  20 C  REM EFF AND DEVICE+CABIN M.GEN MUST BE LOADED INTO MAT DD
  21 C    BEFORE USING THIS SUBROUTINE
  22 C OUTPUTS:
  23 C  M.REM (MG/HR) FOR ALL DEVICES + CABIN CALCULATED AT CAV,
  24 C    ARE STORED IN MAT DD COL 21
  25 C  SMREM=SUM OF MASS OF CONT REM IN DEVICES (MG/HR)
  26 C  SMGEN=SUM OF MASS GENERATED BY ALL DEVICES + CABIN (MG/HR)
  27 C
  28 C  LOAD DEVICES 1 AND 2 WITH CIN AND COUT+M.REMOVED FOR DEV 2
  29 C    DEVICE 1=CABIN
  30 C      LOAD MAT DD WITH CAV CABIN
  31        DD(1,17)=CAV
  32        DD(1,18)=CAV
  33        DD(1,21)=0
  34 C   DEVICE 2
  35        DD(2,17)=CAV
  36        DD(2,18)=CAV
  37        DD(2,21)=DD(2,2)*DD(2,18)
  38 C   FOR DEV 3-15 CALC CIN COUT AND M.REMOVED BY DEVICE
  39 C      IF DEVICE FLOW RATE=0 THEN SET CIN,COUT,& M.GEN=0
  40        DO 100 J=3,LIN2
  41        IF (DD(J,2).EQ.0) THEN
  42          DD(J,17)=0
  43          DD(J,18)=0
  44          DD(J,21)=0
  45          GOTO 100
  46        ENDIF
  47 C  IF UPSTREAM DEVICE=1 0R 2 THEN SET INLET=CABIN CONC+DEV M.GEN/Q
  48          IF((DD(J,4).EQ.1).OR.(DD(J,4).EQ.2)) THEN
  49          DD(J,17)=DD(1,18)+DD(J,19)/DD(J,2)
  50        ELSE
  51 C      DETERMINE FLOWS,CIN AND COUT FOR DEVICES WITH RELATIVE ADDRESSES
  52          IF (DD(J,4).EQ.0) THEN
  53            QNO1=0
  54            CNO1=0
  55          ELSE
  56            QNO1=DD(NINT(DD(J,4)),2)
  57            CNO1=DD(NINT(DD(J,4)),18)
  58          ENDIF
```

```
   59           IF (DD(J,5).EQ.0) THEN
   60             QNO2=0
   61             CNO2=0
   62           ELSE
   63             QNO2=DD(NINT(DD(J,5)),2)
   64             CNO2=DD(NINT(DD(J,5)),18)
   65           ENDIF
   66           IF (DD(J,6).EQ.0) THEN
   67             QNO3=0
   68             CNO3=0
   69           ELSE
   70             QNO3=DD(NINT(DD(J,6)),2)
   71             CNO3=DD(NINT(DD(J,6)),18)
   72           ENDIF
   73 C
   74 C         IF ALL UPSTREAM DEVICE FLOWS=0
   75           IF (QNO1+QNO2+QNO3.EQ.0) THEN
   76             DD(J,17)=0
   77             DD(J,18)=0
   78             DD(J,21)=0
   79             OPEN(2,FILE='CON',IOSTAT=IOVAL)
   80               WRITE(*,*)' FLOW HALTED-UPSTREAM DEV TURNED OFF-
   81      +  INC END TIME;DEV=',TN,DD(J,1)
   82             CLOSE (2)
   83             GO TO 100
   84           ELSE
   85 C           CALCULATE CIN
   86             DD(J,17)=(QNO1*CNO1+QNO2*CNO2+QNO3*CNO3)/
   87      +  (QNO1+QNO2+QNO3)+DD(J,19)/DD(J,2)
   88           ENDIF
   89 C       END OF DETERMINE FLOWS,CIN,COUT OF DEV WITH REL ADDR.
   90         ENDIF
   91 C     CALCULATE COUT
   92          DD(J,18)=DD(J,17)*(1-DD(J,20))
   93 C        CALCULATE SUM OF MASS REMOVED (CIN*Q*REM EFF)
   94          DD(J,21)=DD(J,17)*DD(J,2)*DD(J,20)
   95    100 CONTINUE
   96 C   END OF  LOADING OF MAT DD WITH DATA AND CALCULATING CIN COUT,M.REM
   97 C   SUM TOTAL MASS OF CONT REMOVED BY ALL DEVICES (2-15) (MG/HR)
   98         SMREM=0
   99         DO 101 J=2,LIN2
  100         SMREM=SMREM+DD(J,21)
  101    101 CONTINUE
  102 C   SUM MASS OF CONT GENERATED IN ALL DEVICES+CABIN (1-15) (MG/HR)
  103         SMGEN=0
  104         DO 102 J=1,LIN2
  105         SMGEN=SMGEN+DD(J,19)
  106    102 CONTINUE
  107 C   CALC M.REM CABIN AND PUT IN DD(1,21)
  108         DD(1,21)=SMGEN-SMREM
  109         RETURN
  110 C      ************ END OF SUBROUTINE CALCM ****************************
  111         END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\FORTRAN\TCC\CATBNR Options: /C 80 /L /BIJY   03/15/94 10:39:12

   1
   2 C     ***************************************************************
   3 C     *                    SUBROUTINE  CATBNR                       *
   4 C     *          CALCULATES EFFICIENCY OF CAT BURNER                *
   5 C     ***************************************************************
   6 C
   7         SUBROUTINE CATBNR(P,EMAX,OXNEW,OXID,MW,EFF)
   8 C
   9         REAL MW
  10 C   INPUTS:
  11 C    P=TOTAL POISON (CONTAMINANTS IN NHB CATEGORIES 6, 7 AND 12:
  12 C      CHLOROCARBONS, CHLOROFLUOROCARBONS AND SULFIDES) REMOVED
  13 C      BY CAT BURNER (MG)
  14 C    EMAX=MAXIMUM BED EFFICIENCY (DEC)
  15 C    OXID=DEGREE OF OXIDIZATION OF CHEMICAL (1=FULLY, 0=NONE)
  16 C    MW=MOLECULAR WEIGHT
  17 C   OUTPUTS:
  18 C    EFF=REMOVAL EFF (DEC)
  19 C
  20         IF(OXID.LT.0) OXID=0
  21         IF(OXID.GT.1) OXID=1
  22         EFF=EMAX*OXID
  23 C      EFFICIENCY FOR METHANE (MW=16.04) IS A FUNCTION OF P
  24         IF(MW.EQ.16.04) THEN
  25           OXNEW=OXID
  26           IF(P.LE.5500) THEN
  27             EFF=0.97506*10**(-0.00010507*P)*EMAX*OXID
  28           ELSEIF (P.GT.5500) THEN
  29             EFF=(31.453-1.151*1E-3*P+1.9046*1E-8*P**2
  30      +         -1.0389*1E-13*P**3)*0.01*EMAX*OXID
  31           ENDIF
  32         ELSE
  33           EFF=EMAX*OXID
  34         END IF
  35         IF(EFF.LE.0) EFF=0
  36         IF(EFF.GT.EMAX) EFF=EMAX
  37         RETURN
  38         END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS   IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS   IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\FORTRAN\TCC\CNRSUB Options: /C 80 /L /BIJY   03/15/94 10:39:14

C     ****************************************************************
C     *                SUBROUTINE   CNRSUB                           *
C     *     FOR 1 CONT AT A TIME AT CAV PRED, CALL EFF SUBROUTINES    *
C     *     FOR DEVICES AND PUT EFFICIENCY IN MAT DD COL 20           *
C     ****************************************************************

C     NOTE:DEVICE NUMBERS IN THIS SUBROUTINE REFER TO DEVICE TYPES,
C     NOT THEIR RELATIVE POSITION IN MAT DD

      SUBROUTINE CNRSUB(I,TN,TN1,DD,NROW,NCOL,CC,NROW1,NCOL1,
     +CDI,NROW2,NCOL2,LIN2,KK)
      INTEGER NROW,NCOL,NROW1,NCOL1,NROW2,NCOL2,CATEG,D1,D2,D3,KK
      REAL DD(NROW,NCOL)
      REAL CC(NROW1,NCOL1)
      REAL CDI(NROW2,NCOL2)
      REAL RGTM1,RGTM2,RGTM3,P,POISN,OLDP
C
C SUBROUTINES REQUIRED:
C  ACHBD-REMOVAL EFF OF AXIAL CHARCOAL BED
C  RCHBD-REMOVAL EFF OF RADIAL CHARCOAL BED
C  ALIOH-REMOVAL EFF OF AXIAL LIOH BED
C  COOXID-REMOVAL EFF OF CO OXIDIZER
C CATBNR-REMOVAL EFF OF CAT BURNER
C CONDHX-REMOVAL EFF OF CONDENSING HX
C
C INPUTS:
C FROM PCSET PREDCT,AND CONVRG
C  I=CONT NO.
C  TN,TN1=CONT INCREMENT FINAL,INIT TIME (HRS)
C  CAVPRD=CABIN AVERAGE CONT CONC (MG/CU M)
C  DD,NROW,NCOL=NAME & DIMENSIONS OF MAT DD
C  CC,NROW1,NCOL1=NAME & DIM OF MAT CC
C  CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
C  LIN2=NUMBER OF DEVICES IN MAT DD
C  DD(I,14)=PERCENT RELATIVE HUMIDITY
C FROM EFFICIENCY SUBROUTINES
C  EFF=REMOVAL EFF (DEC)
C OUTPUTS:
C TO EFFICIENCY SUBROUTINES
C  DEVICE AND CONTAMINANT INFORMATION AS REQUIRED
C  DD(J,22)=DEVICE INLET CONCENTRATION
C TO PCSET, PREDCT, & CONVRG
C  PUTS REMOVAL EFF FOR EACH DEVICE IN MAT DD COL 20
C
C EFFICIENCY FOR DEVICES 1 AND 2
      DD(1,20)=0
      DD(2,20)=DD(2,8)

C SET COUNTER FOR READING CHAR SMR IN MAT CC FOR DEVICE TYPE 3
      K=9
C
C BEGIN LOOP FOR DEVICES 3 TO 15 OF MAT DD
      DO 100 J=3,LIN2
C   INDEX COUNTER
      K=K+3
C   SET REM EFF=0 AND GO TO END OF J LOOP IF DEVICE FLOW = 0
      IF(DD(J,2).LE.1E-10) THEN
```

```
58      DD(J,20)=0
59      GOTO 80
60     ENDIF
61  C DECISIONS FOR VARIOUS DEVICES
62     IF (NINT(DD(J,3)).EQ.3) THEN
63  C    GO TO SUBROUTINE FOR DEVICE TYPE 3-AXIAL CHARCOAL BED
64      CALL ACHBD(TN,TN1,DD(J,22),DD(1,10),DD(1,13),DD(J,2),DD(J,8),
65     + DD(J,9),DD(J,10),DD(J,12),DD(J,13),CDI(I,2),CDI(I,3),
66     + CDI(I,4),CDI(I,5),CDI(I,6),CC(I,K),EFF,DD(1,14))
67  C    STORE EFF IN MAT DD
68      DD(J,20)=EFF
69     ELSEIF (NINT(DD(J,3)).EQ.4) THEN
70  C    GO TO SUBROUTINE FOR DEVICE TYPE 4-RADIAL CHARCOAL BED
71      CALL RCHBD(TN,TN1,DD(J,22),DD(1,10),DD(1,13),DD(J,2),DD(J,8),
72     + DD(J,9),DD(J,10),DD(J,11),DD(J,12),DD(J,13),CDI(I,2),CDI(I,3),
73     + CDI(I,4),CDI(I,5),CDI(I,6),CC(I,K),EFF,DD(1,14))
74  C    STORE EFF IN MAT DD
75      DD(J,20)=EFF
76     ELSEIF (NINT(DD(J,3)).EQ.5) THEN
77  C    GO TO SUBROUTINE FOR DEVICE TYPE 5-LIOH BED
78      CALL ALIOH(TN,TN1,DD(J,8),DD(J,9),DD(J,10),DD(J,12),DD(J,15),
79     + DD(J,16),CDI(I,7),EFF)
80  C    STORE EFF IN MAT DD
81      DD(J,20)=EFF
82     ELSEIF (NINT(DD(J,3)).EQ.6) THEN
83  C    GO TO SUBROUTINE FOR DEVICE TYPE 6-CO OXIDIZER
84      CALL COOXID(DD(J,2),DD(J,8),DD(J,9),DD(J,10),CDI(I,4),EFF)
85  C    STORE EFF IN MAT DD
86      DD(J,20)=EFF
87     ELSEIF (NINT(DD(J,3)).EQ.7) THEN
88  C    SUM POISONS (CONTAMINANTS IN NHB CATEGORIES 6, 7 AND 12:
89  C    CHLOROCARBONS, CHLOROFLUOROCARBONS AND SULFIDES) REMOVED
90  C    BY CAT BURNER (MG)
91      P=0.0
92      POISN=0.0
93      DO 200 L=1,NROW1,1
94       CATEG=CDI(L,8)
95        IF((CATEG.EQ.6).OR.(CATEG.EQ.7).OR.(CATEG.EQ.12)) THEN
96          P=CC(L,(J+1)*3)
97          POISN=POISN+P
98        ENDIF
99    200  CONTINUE
100       RGTM1=0
101       RGTM2=0
102       RGTM3=0
103       D1=DD (J,4)
104       D2=DD(J,5)
105       D3=DD)J,6)
106       IF ((D1.NE.0).OR.(D2.NE.0).OR.(D3.NE.0)) THEN
107         IF ((DD(D1,3).EQ.3).OR.(DD(D1,3).EQ.4)) THEN
108           TRCI=DD(D1,15)
109           TRD=DD(D1,16)
110           TIR=DD(D1,14)
111           IF (TN1.EQ.0) GOTO 50
112           IF (TN1.LT.TIR) GOTO 50
113           IF (TRCI.LE.0) GOTO 50
114           IF (AINT((TN1-TIR)/TRCI).EQ.((TN1-TIR)/TRCI)) THEN
115               RGTM1=1
116               GOTO 55
117           ENDIF
```

```
118   050      RGTM1=0
119   055      CONTINUE
120           ENDIF
121           IF ((DD(D2,3).EQ.3).OR.(DD(D2,3).EQ.4)) THEN
122             TRCI=DD(D2,15)
123             TRD=DD(D2,16)
124             TIR=DD(D2,14)
125           IF (TN1.EQ.0) GOTO 60
126           IF (TN1.LT.TIR) GOTO 60
127           IF (TRCI.LE.0) GOTO 60
128           IF (AINT((TN1-TIR)/TRCI).EQ.((TN1-TIR)(/TRCI)) THEN
129               RGTM2=1
130               GOTO 65
131           ENDIF
132   060     RGTM2=0
133   065     CONTINUE
134         ENDIF
135         IF ((DD(D3,3).EQ.3).OR.(DD(D3,3).EQ.4)) THEN
136           TRCI=DD(D3,15)
137           TRD=DD(D3,16)
138           TIR=DD(D3,14)
139           IF (TN1.EQ.0) GOTO 70
140           IF (TN1.LT.TIR) GOTO 70
141           IF (TRCI.LE.0) GOTO 70
142           IF (AINT((TN1-TIR)/TRCI).EQ.((TN1-TIR)/TRCI)) THEN
143               RGTM3=1
144               GOTO 75
145           ENDIF
146   070     RGTM3=0
147   075     CONTINUE
148         ENDIF
149       ENDIF
150       IF((RGTM1.NE.0).OR.(RGTM2.NE.0).OR.(RGTM3.NE.0)) THEN
151         OLDP=POISN
152       REINITIALIZES METHANE OXIDATION EFFICIENCY TO 90% OF PREVIOUS
153         IF ((CDI(I,4).EQ.16.04).AND.(KK.EQ.1)) THEN
154            CDI(I,23)=0.9*CDI(I,23)
155          ENDIF
156       ENDIF
157       POISN=POISN-OLDP
158 C      GO TO SUBROUTINE FOR DEVICE TYPE 7-CAT BURNER
159       CALL CATBNR(POISN,DD(J,8),DD(J,9),CDI(I,23),CDI(I,4),EFF)
160 C      STORE EFF IN MAT DD
161       DD(J,20)=EFF
162      ELSEIF (NINT(DD(J,3)).EQ.8) THEN
163 C      GO TO SUBROUTINE FOR DEVICE TYPE 8-CONDENSING HX
164       CALL CONDHX(DD(J,2),DD(J,8),DD(J,9),CDI(I,4),CDI(I,5),CDI(I,6),
165     + DD(J,22),EFF)
166 C      STORE EFF IN MAT DD
167       DD(J,20)=EFF
168      ELSEIF (NINT(DD(J,3)).EQ.9) THEN
169 C      SUBROUTINE FOR DEVICE TYPE 9-DUMMY
170       EFF=0
171 C      STORE EFF IN MAT DD
172       DD(J,20)=EFF
173      ELSE
174 C      DEVICES >9 OR <1 (OR ANY DEVICE NOT IN ABOVE CASES)
175       DD(J,20)=0
176      ENDIF
177    080 CONTINUE
178    100 CONTINUE
179      RETURN
180 C     ********** END OF SUBROUTINE CNRSUB ****************************
181      END
```

```
NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS  IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\CONDHX. Options: /C 80 /L /BY   05/21/92 12:55:36
    1 C      ***************************************************************
    2 C      *                    SUBROUTINE CONDHX                        *
    3 C      *         CALCULATES EFFICIENCY OF CONDENSING HX              *
    4 C      ***************************************************************
    5 C
    6        SUBROUTINE CONDHX(BEDQ,EMAX,MLIQ,MW,VCONC,SOL,CAVPRD,EFF)
    7        REAL MW,MLIQ,H,PA,XA,NOUT
    8 C
    9 C INPUTS:
   10 C    BEDQ=BED FLOW RATE (M3/HR)
   11 C    EMAX=MAXIMUM POSSIBLE REMOVAL EFFICIENCY (DEC)
   12 C    MLIQ=WATER FLOW RATE IN HX DUE TO CONDENSING (KG/HR)
   13 C    MW=MOLECULAR WEIGHT OF CONTAMINANT
   14 C    VCONT=VAPOR CONCENTRATION OF CONTAMINANT (MG/M3)
   15 C    SOL=HENRY'S LAW COEFFICIENT (ATM/MOL FRACTION)
   16 C OUTPUTS:
   17 C    EFF=REMOVAL EFF (DEC)
   18 C
   19         CAIN=CAVPRD
   20         IF (CAIN.LE.1E-10) THEN
   21          CAIN=0.1E-10
   22         ENDIF
   23         IF (SOL.LE.1E-10) THEN
   24          EFF=0
   25         ELSE
   26 C    IF CONTAMINANT IS AMMONIA - USES EXPERIMENTAL DATA FROM JSC-08797
   27 C     FOR AMMONIA REMOVAL AS A FUNCTION OF CO2 CONCENTRATION (8/23/76)
   28           IF (MW.EQ.17.0) THEN
   29               CAOUT=((CAIN*BEDQ)-(MLIQ*189.5847418*CAIN**0.534915256))/BEDQ
   30               EFF=((CAIN-CAOUT)/CAIN)*EMAX
   31           ELSE
   32 C    CONTAMINANT IS NOT AMMONIA
   33 C    CALCULATE CONTAMINANT PARTIAL PRESSURE AND WATER MOLE FRACTION
   34               PA=CAIN*1.0E-9*82.06*278/MW
   35               XA=(PA/1)/((MLIQ*(1000/18))/(BEDQ*(1000/22.4))+SOL/1)
   36               NOUT=MLIQ*XA*1000/18
   37               CAOUT=((CAIN*BEDQ)-(NOUT*MW*1000))/BEDQ
   38               EFF=((CAIN-CAOUT)/CAIN)*EMAX
   39           ENDIF
   40         ENDIF
   41         IF(EFF.LE.0) EFF=0
   42         IF(EFF.GT.EMAX) EFF=EMAX
   43         RETURN
   44         END
   45 C       ***************** END OF SUBROUTINE CONDHX **********************

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS  IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\CONVRG. Options: /C 80 /L /BY   05/21/92 12:55:41
C      ****************************************************************
C      *                                                              *
C      *           SUBROUTINE CONVRG                                  *
C      *       MAIN CONVERGENCE LOOP SUBROUTINE                       *
C      * USING CAV PRED & BASED ON SUM MASS REM OF LAST INCR, CALC     *
C      * NEW REM EFF, CAV CALC, CEQULIB, CFINAL, & M. REMOVED         *
C      * PUT THEM IN MAT DD - WORKS FOR ONE CONT AT A TIME            *
C      ****************************************************************

       SUBROUTINE CONVRG(I,TN,TN1,CAVPRD,DD,NROW,NCOL,
      +CC,NROW1,NCOL1,CDI,NROW2,NCOL2,CAVCLC,CFINAL,CEQLIB,KK,LIN,LIN2,
      +NN,PRTSW3,IMSGDN)
       INTEGER NROW,NCOL,NROW1,NCOL1,NROW2,NCOL2,PRTSW3,KK
       CHARACTER NN(NROW1)*30
       REAL DD(NROW,NCOL)
       REAL CC(NROW1,NCOL1)
       REAL CDI(NROW2,NCOL2)
       REAL X1,X2,Y1,Y2,SLOPE

C SUBROUTINES REQUIRED:
C PRAFIL-ZERO MAT DD COL 17-21
C CNRSUB-USING CAV PRED CALC-REM EFF FOR ALL DEVICES-PUT IN DD COL 20
C MASBAL-CALC CAV CALC,CFINAL,CEQULIB,M.REM
C INPUTS:
C FROM MCALC
C      I=CONTAMINANT NO.
C      TN,TN1 =INCREMENT END & BEGINNING TIME (HRS)
C      DD,NROW,NCOL=NAME & DIM OF MAT DD
C      CC,NROW1,NCOL1=NAME & DIM OF MAT CC
C      CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
C      LIN=NO. OF CONTAMINANTS IN MAT CDI
C      LIN2=NO. DEVICES IN MAT DD
C      IMSGDN=DEVICE NO. FOR MESSAGE OUTPUT
C      CAVPRD=PREDICTED CABIN CONC FOR INCREMENT (MG/CU M)
C FROM PRAFIL
C      ZEROS IN MAT DD COL 17-21
C FROM CNRSUB
C      CALL REM EFF SUBROUTINE & PUTS REM EFF FOR EACH DEV IN DD COL 20
C FROM MASBAL
C      CAVCLC=CALC CABIN CONT CONC (MG/CU M)
C      CFINAL=FINAL INCR CABIN CONT CONC(MG/CUM)
C      CEQLIB=EQUILIBRIUM CABIN CONT CONC(MG/CU M)
C OUTPUTS:
C TO MCALC
C      CAVCLC=CALC CABIN CONT CONC (MG/CU M)
C      CFINAL=FINAL INCR CABIN CONT CONC(MG/CUM)
C      KK=COUNTER FOR CONVERGENCE
C      CEQLIB=EQUILIBRIUM CABIN CONT CONC(MG/CU M)
C      M.REM IS IN MAT DD COL 21
C TO PRAFIL
C      NAME & SIZE OF MATRIX + FIRST AND LAST COL TO BE ZEROED
C TO CNRSUB
C      I=CONT NO.
C      TN,TN1=CONT INCREMENT FINAL,INIT TIME (HRS)
C      DEVICE AVERAGE CONT CONC (MG/M3) = DD(J,22)
C      DD,NROW,NCOL=NAME & DIMENSIONS OF MAT DD
C      CC,NROW1,NCOL1=NAME & DIM OF MAT CC
C      CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
C      LIN2=NUMBER OF DEVICES IN MAT DD
```

```
 59 C TO MASBAL
 60 C       I
 61 C       TN,TN1
 62 C       CVOL=CABIN VOL (CU M)=DD(1,9)
 63 C       CINIT=INCR INIT CABIN CONT CONC (MG/CU M)=CC(I,1)
 64 C
 65 C       CONVERGENCE ERROR (DEC)
 66           CNVERR=DD(1,12)
 67           IF(CNVERR.LT.1E-10) THEN
 68             OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
 69              WRITE(IMSGDN,*) 'CONV ERROR<1E-10:PROGRAM TERMINATED'
 70             CLOSE(IMSGDN)
 71             STOP
 72           ENDIF
 73           KK=1
 74           DO 100 KK=1,20
 75 C           ZERO MAT DD COL 17-21
 76             CALL PRAFIL(DD,NROW,NCOL,17,21)
 77 C           USING CAVPRD FIND REM EFF OF EACH DEV & PUT IN DD COL 20
 78             CALL CNRSUB(I,TN,TN1,DD,NROW,NCOL,CC,NROW1,NCOL1,
 79      +   CDI,NROW2,NCOL2,LIN2,KK)
 80 C           FIND CAVCLC FOR THESE REMOVAL EFFICIENCIES
 81             CALL MASBAL(I,TN,TN1,DD,NROW,NCOL,CC,NROW1,NCOL1,
 82      +    CAVCLC,CDI,NROW2,NCOL2,CFINAL,CEQLIB,LIN,LIN2)
 83 C         IF PRTSW3=1 THEN PRINT NAME & NO + CONV VALUES
 84             IF (PRTSW3.EQ.1) THEN
 85              OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
 86               WRITE(IMSGDN,*)'PRINTOUT FOR CONVERGENCE VALUES IN CONVRG'
 87               WRITE(IMSGDN,50) I,NN(I)
 88  050          FORMAT(1X,'CONT NO.= ',I4,2X,A)
 89               WRITE(IMSGDN,*)'CAVPRD,CAVCLC= ',CAVPRD,CAVCLC
 90              CLOSE(IMSGDN)
 91             ENDIF
 92 C
 93 C         IF CAVCLC=CAVPRD THEN EXIT THE KK LOOP
 94             IF(CAVCLC.EQ.CAVPRD) GOTO 101
 95 C         IF CAVPRD<1E-10 THEN SKIP CONVERGENCE STEP
 96             IF(CAVPRD.LT.1E-10) GOTO 80
 97 C         IF CONVERGENCE IS REACHED EXIT THE KK LOOP
 98             IF(ABS((CAVPRD-CAVCLC)/CAVPRD).LT.CNVERR) THEN
 99              GOTO 101
100             ENDIF
101 C       CONVERGENCE CALCULATION ROUTINE
102 C       USE THE BISECTION METHOD FOR THE ITERATION WHERE KK=1
103  080   IF (KK.EQ.1) THEN
104 C       INITIALIZE X2 AND Y2 FOR THE NEXT ITERATION
105             X2=CAVPRD
106             Y2=CAVCLC-CAVPRD
107             CAVPRD=(CAVPRD+CAVCLC)/2
108           ELSE
109 C       USE THE NEWTON-RAPHSON METHOD FOR ITERATIONS WHERE KK>1
110             X1=X2
111             Y1=Y2
112             X2=CAVPRD
113             Y2=CAVCLC-CAVPRD
114             SLOPE=(Y2-Y1)/(X2-X1)
115             CAVPRD=X2-0.95*Y2/SLOPE
116           ENDIF
117 C
118 C       SET CAV IN PRED DD(I,22) = CAV IN CALC DD(I,17)
```

```
119         DO 90 J=1,LIN2
120           DD(J,22)=DD(J,17)
121   090 CONTINUE
122 C
123   100 CONTINUE
124 C       END OF KK LOOP
125   101 CONTINUE
126 C
127
128         RETURN
129 C    ************ END OF SUBROUTINE CONVRG  ****************************
130         END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS  IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\COOXID. Options: /C 80 /L /BY   05/21/92 12:55:52
    1 C      ****************************************************************
    2 C      *          SUBROUTINE COOXID                                   *
    3 C      *  CALCULATES EFFICIENCY OF CO OXIDIZER (Pt on charcoal)       *
    4 C      ****************************************************************
    5 C
    6        SUBROUTINE COOXID(BEDQ,EMAX,BEDL,BEDDIA,MW,EFF)
    7        REAL MW
    8 C
    9 C INPUTS:
   10 C    BEDQ=BED FLOW RATE (CU M/HR)
   11 C    EMAX=MAXIMUM POSSIBLE REMOVAL EFFICIENCY (DEC)
   12 C    BEDL=BED LENGTH (M)
   13 C    BEDDIA=BED DIAMETER (M)
   14 C    MW=MOLECULAR WEIGHT OF CONTAMINANT
   15 C OUTPUTS:
   16 C    EFF=REMOVAL EFF (DEC)
   17 C
   18 C      WORKS ONLY FOR C0 MW=28.01 OR H2=2.02; OTHERWISE REM EFF=0
   19        IF ((MW.EQ.28.01).OR.(MW.EQ.2.02)) THEN
   20         EFF=EMAX
   21 C      IF RESIDENCE TIME <0.2 SEC THEN REM EFF DROPS LINEARLY
   22 C      BREST = BED RESIDENCE TIME (SEC)
   23         BREST=(3.141592654/4)*BEDL*BEDDIA**2*3600/BEDQ
   24         IF (BREST.LT.0.2) THEN
   25          EFF=EMAX*BREST/0.2
   26         ENDIF
   27        ELSE
   28 C       REM EFF FOR OTHER THAN CO OR H2 = ZERO
   29         EFF=0
   30        ENDIF
   31        IF(EFF.LE.0) EFF=0
   32        IF(EFF.GT.EMAX) EFF=EMAX
   33        RETURN
   34        END
   35 C      ***************** END OF SUBROUTINE COOXID ***********************

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS  IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\CRIN.FO Options: /C 80 /L /BY   05/21/92 12:55:55
   1 C      ************************************************************
   2 C      *     SUBROUTINE  CRIN                                     *
   3 C      *   SUBROUTINE TO READ STRING OF LENGTH 30 INTO MAT NN     *
   4 C      *   AND READ REAL DATA INTO MAT XX(ROW,COL)                *
   5 C      *   RETURNS NUMBER OF LINES OF DATA READ FROM FILE         *
   6 C      ************************************************************
   7 C    NOTE: INPUT STRING MUST HAVE SINGLE QUOTES AROUND IT
   8 C    NOTE: INPUT NUMBERS MUST BE SEPARATED BY BLANKS
   9        SUBROUTINE CRIN(NN,XX,NROW,NCOL,LIN)
  10        INTEGER NROW,NCOL,IOVAL,LIN
  11        CHARACTER NN(NROW)*30,FNAME*24
  12        REAL XX(NROW,NCOL)
  13  010 READ(*,'(A)') FNAME
  14        OPEN(1,FILE=FNAME,STATUS='OLD',IOSTAT=IOVAL)
  15        IF(IOVAL.NE.0) GOTO 900
  16        LIN=0
  17        DO 100 I=1,NROW
  18        READ(1,*,IOSTAT=IOVAL,END=500,ERR=900 ) NN(I),(XX(I,J),J=1,NCOL)
  19        LIN=LIN+1
  20  100 CONTINUE
  21  500 WRITE(*,'(A)') ' DONE WITH FILE INPUT'
  22        WRITE (*,*) ' '
  23        CLOSE (1)
  24        GOTO 990
  25  900 WRITE(*,*)'IOERROR= ',IOVAL
  26        CLOSE (1)
  27        WRITE(*,*) 'WHAT IS THE INPUT FILE NAME? '
  28        GOTO 10
  29  990 RETURN
  30 C      *************** END OF SUBROUTINE CRIN ************************
  31        END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS  IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\CROUT.F Options: /C 80 /L /BY   05/21/92 12:56:01
    1 C      ************************************************************
    2 C      *    SUBROUTINE CROUT                                      *
    3 C      *  SUBROUTINE TO WRITE DATA TO CONSOLE, OR PRINTER         *
    4 C      * WRITES STRG OF LENGTH 30 FROM MAT NN & REAL DATA FROM MAT *
    5 C      * XX(ROW,COL) STARTING WITH COL FSTCOL, AND ENDING WITH LSTCOL *
    6 C      *  AND FROM LINE FSTLIN TO LINE LSTLIN                     *
    7 C      ************************************************************
    8        SUBROUTINE CROUT(NN,XX,NROW,NCOL,FSTCOL,LSTCOL,LIN,FSTLIN,LSTLIN,
    9     +IMSGDN,NINC,FNAME,IDEVNO,IOVAL)
   10        INTEGER
NROW,NCOL,IOVAL,FSTCOL,LSTCOL,LIN,FSTLIN,LSTLIN,NINC,IDEVNO,
   11     + IOVAL
   12        CHARACTER FNAME*24,DES*1
   13        CHARACTER NN(NROW)*30
   14        REAL XX(NROW,NCOL)
   15        IF (FSTCOL.GT.NCOL) FSTCOL=NCOL
   16        IF (LSTCOL.GT.NCOL) LSTCOL=NCOL
   17        IF (FSTCOL.GT.LSTCOL) FSTCOL=LSTCOL
   18        IF (FSTLIN.GT.LIN) FSTLIN=LIN
   19        IF (LSTLIN.GT.LIN) LSTLIN=LIN
   20        IF (FSTLIN.GT.LSTLIN) FSTLIN=LSTLIN
   21
   22 C 010 OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
   23 C      WRITE(IMSGDN,'(A)') ' WRITE TO LPT1 OR CON OR END '
   24 C      CLOSE (IMSGDN)
   25 C      READ(*,'(A)') FNAME
   26 C      QUIT IF FNAME=END
   27 C      IF (FNAME.EQ.'END') GO TO 990
   28 C      IF((FNAME.NE.'LPT1').AND.(FNAME.NE.'CON')) GOTO 10
   29 C      OPEN(1,FILE=FNAME, IOSTAT=IOVAL)
   30        IF(IOVAL.NE.0) GOTO 900
   31          WRITE (IDEVNO,55,IOSTAT=IOVAL,ERR=900) NINC
   32  055   FORMAT ('INCREMENT NO. = ',I7)
   33        DO 110 I=FSTLIN,LSTLIN
   34 C        WRITE(1,60,IOSTAT=IOVAL,ERR=900) I,NN(I)
   35          WRITE(IDEVNO,60,IOSTAT=IOVAL,ERR=900) I,NN(I)
   36  060  FORMAT(1X,'CONT NO.= ',I4,2X,A)
   37 C        WRITE(1,70,IOSTAT=IOVAL,ERR=900) (XX(I,J),J=FSTCOL,LSTCOL)
   38          WRITE(IDEVNO,70,IOSTAT=IOVAL,ERR=900) (XX(I,J),J=FSTCOL,LSTCOL)
   39  070  FORMAT(1X,7G11.4)
   40  110 CONTINUE
   41 C      CLOSE (1)
   42        GOTO 990
   43  900 OPEN (IMSGDN,FILE='CON',IOSTAT=IOVAL)
   44        WRITE(IMSGDN,*)'IOERROR= ',IOVAL
   45        CLOSE(IMSGDN)
   46 C      CLOSE (1)
   47        CLOSE (IDEVNO)
   48 C      GOTO 10
   49  990 RETURN
   50 C      ********** END OF SUBROUTINE CROUT ******************************
   51        END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS  IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\CROUT2. Options: /C 80 /L /BY   05/21/92 12:56:09
   1 C      ***************************************************************
   2 C      *                                                             *
   3 C      *    SUBROUTINE CROUT2                                        *
   4 C      *  SUBROUTINE TO WRITE DATA TO CONSOLE, OR PRINTER            *
   5 C      * WRITES STRG OF LENGTH 30 FROM MAT NN & REAL DATA FROM MAT    *
   6 C      * XX(ROW,COL) STARTING WITH COL FSTCOL, AND ENDING WITH LSTCOL *
   7 C      *  AND FROM LINE FSTLIN TO LINE LSTLIN                        *
   8 C      ***************************************************************
   9        SUBROUTINE CROUT2(NN,XX,NROW,NCOL,FSTCOL,LSTCOL,LIN,FSTLIN,LSTLIN,
  10     +IMSGDN)
  11        INTEGER NROW,NCOL,IOVAL,FSTCOL,LSTCOL,LIN,FSTLIN,LSTLIN
  12        CHARACTER FNAME*24,DES*1
  13        CHARACTER NN(NROW)*30
  14        REAL XX(NROW,NCOL)
  15        IF (FSTCOL.GT.NCOL) FSTCOL=NCOL
  16        IF (LSTCOL.GT.NCOL) LSTCOL=NCOL
  17        IF (FSTCOL.GT.LSTCOL) FSTCOL=LSTCOL
  18        IF (FSTLIN.GT.LIN) FSTLIN=LIN
  19        IF (LSTLIN.GT.LIN) LSTLIN=LIN
  20        IF (FSTLIN.GT.LSTLIN) FSTLIN=LSTLIN
  21
  22   010 OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
  23        WRITE(IMSGDN,'(A)') ' WRITE TO LPT1 OR CON OR END '
  24        CLOSE (IMSGDN)
  25        READ(*,'(A)') FNAME
  26        QUIT IF FNAME=END
  27        IF (FNAME.EQ.'END') GO TO 990
  28        IF((FNAME.NE.'LPT1').AND.(FNAME.NE.'CON')) GOTO 10
  29        OPEN(1,FILE=FNAME, IOSTAT=IOVAL)
  30        IF(IOVAL.NE.0) GOTO 900
  31        DO 110 I=FSTLIN,LSTLIN
  32         WRITE(1,70,IOSTAT=IOVAL,ERR=900) (XX(I,J),J=FSTCOL,LSTCOL)
  33   070  FORMAT(1X,7G11.4)
  34   110 CONTINUE
  35        CLOSE (1)
  36        GOTO 990
  37   900 OPEN (IMSGDN,FILE='CON',IOSTAT=IOVAL)
  38        WRITE(IMSGDN,*)'IOERROR= ',IOVAL
  39        CLOSE(IMSGDN)
  40        CLOSE (1)
  41        GOTO 10
  42   990 RETURN
  43 C      ********** END OF SUBROUTINE CROUT ******************************
  44        END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS  IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\DATOUT. Options: /C 80 /L /BY   05/21/92 12:56:16
    1 C      ****************************************************************
    2 C      *    SUBROUTINE DATOUT                                         *
    3 C      * SUBROUTINE TO PRINT HEADINGS AND DATA TO PRINTER,CON,OR FILE *
    4 C      ****************************************************************
    5 C      NOTES:(1)FILE MUST BE OPEN BEFORE CALLING THIS SUBROUTINE
    6 C         (2) IDEVNO MUST BE 6 FOR FORM FEEDS TO BE PRINTED
    7
    8        SUBROUTINE DATOUT(TN,TN1,LIN,DD,NROW,NCOL,CC,NROW1,NCOL1,
    9       +CDI,NROW2,NCOL2,LIN2,NN,PRTSW6,PRTSW8,PRTSW9,
   10        +IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR,
   11       +TVAL,FCPLOT,IDEVN1,IDEVN3,IDEVN2)
   12
   13        INTEGER PRTSW6,PRTSW8,PRTSW9,TVAL,NINC
   14
   15 C  SUBROUTINES REQUIRED:
   16 C       PRCDA=PRINT OUT OF CONTAMINANT DATA
   17 C       PRREM1=PRINTOUT 0F RATE OF CONTAMINANT REMOVAL (MG/HR)-SHEET1
   18 C       PRREM2=PRINTOUT 0F RATE OF CONTAMINANT REMOVAL (MG/HR)-SHEET2
   19 C       PRMAS1=PRINTOUT OF SUM MASSES REMOVED BY DEVICES (MG)-SHEET1
   20 C       PRMAS2=PRINTOUT OF SUM MASSES REMOVED BY DEVICES (MG)-SHEET2
   21 C       PREFF=PRINTOUT OF INCREMENT END REMOVAL EFFICIENCIES
   22
   23 C  INPUTS FROM MAIN PROGRAM:
   24 C       TN=INCREMENT FINAL TIME (HRS)
   25 C       TN1=INCREMENT INITIAL TIME (HRS)
   26 C       LIN=NO. OF CONT IN MAT CC A D NN
   27 C       DD,NROW,NCOL=NAME & SIZE OF MAT DD
   28 C       CC,NROW1,NCOL1=NAME & SIZE F MAT CC
   29 C       CDI,NROW2,NCOL2=NAME & SIZE OF MAT CDI
   30 C       LIN2=NO. DEVICES IN MAT DD
   31 C       NN=NAME OF MAT NN
   32 C       NINC=TIME INCREMENT NUMBER
   33 C         =0 THEN PRINT HEADINGS & DATA FOR PRECALCULATION SET UP ROUTINE
   34 C         =-1 THEN PRINT HEADINGS & DATA FOR FINAL ANSWERS
   35 C         ELSE PRINT WITH PROPER INCREMENT NUMBER
   36 C       IDEVNO=OUTPUT DEVICE NUMBER (SHOULD BE 6)
   37 C       IMONTH..IHOUR=DATE AND TIME VARIABLES
   38 C       IPGCTR=PAGE COUNTER FOR SEQUENTIAL PAGE NO.'S ON ALL PAGES
   39
   40 C  OUTPUT TO MAIN PROG:
   41 C       IOVAL=STATUS OF IOERROR IN SUBROUTINES
   42
   43 * TEST CASE **********
   44 *       LIN=120
   45
   46 C       PRINT OUT CONCENTRATION DATA
   47         CALL PRCDA(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
   48        + IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR,
   49        + PRTSW8,PRTSW9,FCPLOT,IDEVN1)
   50 C       PRINT OUT NHB 8060.1 GROUP CONTRIBUTION VALUES (T-VALUES)
   51         IF ((TVAL.EQ.1).OR.(TVAL.EQ.2)) THEN
   52         CALL GROUP(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
   53        + IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,
   54        + IPGCTR,TVAL,IDEVN3,PRTSW8)
   55         ENDIF
   56         IF (PRTSW6.EQ.1) THEN
   57          IF ((PRTSW8.EQ.1).OR.((PRTSW8.EQ.0).AND.(NINC.EQ.-1))) THEN
   58 C         PRINTOUT OF RATE OF CONTAMINANT REMOVAL BY DEVICES-SHEET1
```

```
 59           CALL PRREM1(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
 60          + IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR)
 61 C         DON'T PRINT SHEET 2 UNLESS NUMBER OF DEVICES IN MAT DD > 8
 62           IF (LIN2.GT.8) THEN
 63 C          PRINTOUT OF RATE OF CONTAMINANT REMOVAL BY DEVICES-SHEET2
 64            CALL PRREM2(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
 65          +  IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR)
 66           ENDIF
 67 C         PRINTOUT OF SUM OF MASS REMOVED BY DEVICES-SHEET1
 68           CALL PRMAS1(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
 69          + IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR)
 70 C         DON'T PRINT SHEET 2 UNLESS NUMBER OF DEVICES IN MAT DD > 8
 71           IF(LIN2.GT.8) THEN
 72 C          PRINTOUT OF SUM OF MASS REMOVED BY DEVICES-SHEET2
 73            CALL PRMAS2(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
 74          +  IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR)
 75           ENDIF
 76          ENDIF
 77 C         PRINTOUT OF INCREMENT END REMOVAL EFFICIENCIES
 78           CALL PREFF(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
 79          + IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,
 80          + IPGCTR,PRTSW8,PRTSW9,IDEVN2)
 81          ENDIF
 82
 83          RETURN
 84          END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS  IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\GROUP.F Options: /C 80 /L /BY   05/21/92 12:56:54
   1 C FILE GROUP
   2 C      ***********************************************************************
   3 C      *    SUBROUTINE GROUP                                                 *
   4 C      * PROGRAM TO PRINT THE GROUP TOXICITY LEVELS AND T LEVEL              *
   5 C      ***********************************************************************
   6 C
   7 C
   8         SUBROUTINE GROUP(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
   9        +IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,
  10        +IPGCTR,TVAL,IDEVNT,PRTSW8)
  11 C
  12 C
  13         DIMENSION GL(16)
  14         REAL CC(NROW1,NCOL1)
  15         REAL CDI(NROW2,NCOL2)
  16         REAL TLEVL
  17         INTEGER TVAL,PRTSW8,NINC,IDEVNO,IDEVNT
  18 C       DETERMINE THE SUMS FOR EACH GROUP LEVEL
  19         DO 25 J=1,16
  20   25 GL(J)=0.
  21         DO 30 I=1,LIN
  22           FRACT = CC(I,4)/CDI(I,9)
  23           TSTR = CDI(I,8)
  24           NHB = IFIX(TSTR)
  25           GL(NHB) = GL(NHB) + FRACT
  26   30 CONTINUE
  27 C       CALCULATE THE TLEVEL OF THE ASSOCIATED GROUP LEVELS
  28         TLEVL = GL(1)+GL(2)+GL(3)+GL(4)+GL(5)+GL(9)+GL(10)+GL(11)+
  29        +     GL(13)+GL(14)+GL(16)
  30         IF ((PRTSW8.EQ.1).OR.((PRTSW8.EQ.0).AND.(NINC.EQ.-1))) THEN
  31         WRITE (IDEVNO,*,IOSTAT=IOVAL,ERR=900) ' '
  32         WRITE (IDEVNO,*,IOSTAT=IOVAL,ERR=900) ' '
  33         WRITE (IDEVNO,*,IOSTAT=IOVAL,ERR=900)
  34        +' GROUP T-VALUES AS SPECIFIED IN NHB 8060.1B APPENDIX D'
  35         WRITE (IDEVNO,*,IOSTAT=IOVAL,ERR=900) ' '
  36         WRITE (IDEVNO,*,IOSTAT=IOVAL,ERR=900)
  37        +' -01-  -02-  -03-  -04-  -05-  -06-  -07-  -08-  -09-
  38        +-10-  -11-  -12-  -13-  -14-  -15-  -16-'
  39         WRITE (IDEVNO,'(/1X,16(F6.2,1X)/)',IOSTAT=IOVAL,ERR=900) GL(1),
  40        + GL(2),GL(3),GL(4),GL(5),GL(6),GL(7),GL(8),GL(9),GL(10),
  41        + GL(11),GL(12),GL(13),GL(14),GL(15),GL(16)
  42         WRITE (IDEVNO,*,IOSTAT=IOVAL,ERR=900) ' OVERALL T-VALUE'
  43         OALLT = GL(1)+GL(2)+GL(3)+GL(4)+GL(5)+GL(9)+GL(10)+GL(11)+
  44        +     GL(13)+GL(14)+GL(16)
  45         WRITE (IDEVNO,'(T4,F7.2)',IOSTAT=IOVAL,ERR=900) OALLT
  46         ENDIF
  47         IF (NINC.NE.-1) THEN
  48 C       ************* WRITE T-VALUE DATA TO A PLOT FILE ****************
  49          IF (TVAL.EQ.2) THEN
  50           WRITE (IDEVNT,50,IOSTAT=IOVAL,ERR=900) TN1,TN,TLEVL
  51    50     FORMAT (T2,2(F8.2,1X),F7.2)
  52          ENDIF
  53         ENDIF
  54         GOTO 999
  55 C       ********************** END OF SUBROUTINE **********************
  56   900 WRITE(*,*)'IO ERROR IN GROUP= ',IOVAL
  57   999 RETURN
  58         END
```

```
NUMBER OF WARNINGS IN PROGRAM UNIT:  0
NUMBER OF ERRORS  IN PROGRAM UNIT:  0
NUMBER OF WARNINGS IN COMPILATION :  0
NUMBER OF ERRORS  IN COMPILATION :  0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\FORTRAN\TCC\HEADGS Options: /C 80 /L /BIJY  03/17/94 09:59:34

  1 C FILE:HEADGS.FOR
  2 C   ***************************************************************
  3 C   *    SUBROUTINE HDG1                                          *
  4 C   * PROGRAM TO PRINT HEADING-DATE, TIME, FILE NAME, & PAGE NO.  *
  5 C   ***************************************************************
  6
  7 C   NOTE:FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
  8
  9     SUBROUTINE HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,
 10    +FNAME,IPGNO,IDEVNO)
 11 C   IMONTH...ISECOND=TIME AND DATE NAMES
 12 C   FNAME=FILE NAME
 13 C   IPGNO=PAGE NUMBER
 14 C   IDEVNO=DEVICE NUMBER FOR OUTPUT
 15
 16     CHARACTER FNAME*24
 17
 18
 19 C   WRITE HEADING
 20     IF (IPGNO .EQ. 1) THEN
 21     WRITE (IDEVNO,5,IOSTAT=IOVAL,ERR=900)
 22   005  FORMAT (2X,'PROGRAM VERSION 8.1 Alpha',5X,'March 15, 1994',/)
 23     ENDIF
 24     WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900) IMONTH,IDAY,IYEAR,
 25    +IHOUR,IMINUTE,FNAME,IPGNO
 26   010 FORMAT(2X,I2,'/',I2,'/',I4,5X,I2,':',I2,5X,A,2X,'PAGE ',I4)
 27
 28     GO TO 999
 29   900 WRITE(*,*)'IO ERROR IN HDG1= ',IOVAL
 30   999 RETURN
 31     END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0

 32
 33 C   ***************************************************************
 34 C   *    SUBROUTINE DATTM                                         *
 35 C   * SUBROUTINE TO READ SYSTEM DATE AND TIME FOR IBM PC OR AT     *
 36 C   ***************************************************************
 37
 38     SUBROUTINE DATTM(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,ISECOND)
 39
 40 C    REQUIRED FOR IBM PROF FORTRAN
 41     INTEGER*2 IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,ISECOND,IHUNSEC
 42
 43     CALL GETDAT(IYEAR,IMONTH,IDAY)
 44     CALL GETTIM(IHOUR,IMINUTE,ISECOND,IHUNSEC)
 45
 46     RETURN
 47     END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0
```

```
48
49 C    ***************************************************************
50 C    *     SUBROUTINE HDG2                                         *
51 C    * PROGRAM TO PRINT HEADING-TIME INCR+INCR INIT AND FINAL TIME  *
52 C    * PRINTS TIME INCREMENT NUMBER                                *
53 C    ***************************************************************
54
55 C    NOTE:FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
56
57     SUBROUTINE HDG2(INCRNO,TN1,TN,IDEVNO)
58
59 C   INPUTS:
60 C    INCRNO=TIME INCREMENT NUMBER
61 C    TN1=INCREMENT INITIAL TIME (HRS)
62 C    TN=INCREMENT FINAL TIME (HRS)
63 C    IDEVNO=DEVICE NUMBER FOR OUTPUT
64
65 C    WRITE HEADING
66     WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900)INCRNO,TN1,TN
67   010 FORMAT(1X,'TIME INCR ',I5,2X,'INITIAL TIME (HRS)= ',F8.2,2X,
68    +'FINAL TIME (HRS)= ',F8.2)
69
70     GO TO 999
71   900 WRITE(*,*)'IO ERROR IN HDG2= ',IOVAL
72   999 RETURN
73     END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0

74
75 C    ***************************************************************
76 C    *     SUBROUTINE HDG3                                         *
77 C    * PROGRAM TO PRINT HEADING-TIME INCR+INCR INIT AND FINAL TIME  *
78 C    * PRINTS PCALC OR FINAL INSTEAD OF TIME INCREMENT NUMBER       *
79 C    ***************************************************************
80
81 C    NOTE:FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
82
83     SUBROUTINE HDG3(IFLAG,TN1,TN,IDEVNO)
84
85 C   INPUTS:
86 C    IFLAG=FLAG FOR TIME INCREMENT (1=PCALC, 2=FINAL)
87 C    TN1=INCREMENT INITIAL TIME (HRS)
88 C    TN=INCREMENT FINAL TIME (HRS)
89 C    IDEVNO=DEVICE NUMBER FOR OUTPUT
90
91     CHARACTER INAME*5
92
93     IF(IFLAG.EQ.1) THEN
94      INAME='PCALC'
95     ELSEIF(IFLAG.EQ.2) THEN
96      INAME='FINAL'
97     ELSE
98      INAME='ERROR'
99     ENDIF
100
101 C    WRITE HEADING
102     WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900)INAME,TN1,TN
103   010 FORMAT(1X,'TIME INCR ',A,2X,'INITIAL TIME (HRS)= ',F8.2,2X,
```

```
  104    +'FINAL TIME (HRS)= ',F8.2)
  105
  106     GO TO 999
  107   900 WRITE(*,*)'IO ERROR IN HDG3= ',IOVAL
  108   999 RETURN
  109     END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0

  110
  111 C    ***************************************************************
  112 C    *     SUBROUTINE HDG4                                         *
  113 C    * PROGRAM TO PRINT HEADING-CONT NO., NAME, FINAL CABIN CONC    *
  114 C    * MAC, EXCEEDS MAC                                            *
  115 C    ***************************************************************
  116
  117 C    NOTE:FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
  118
  119     SUBROUTINE HDG4(IDEVNO)
  120
  121 C   INPUTS:
  122 C    IDEVNO=DEVICE NUMBER FOR OUTPUT
  123
  124 C    WRITE HEADING
  125     WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900)
  126   010 FORMAT(1X,'CONT',14X,'NAME',14X,'FINAL CABIN',5X,'MAC',5X,
  127    +'EXCEEDS')
  128     WRITE(IDEVNO,20,IOSTAT=IOERR,ERR=900)
  129   020 FORMAT(1X,' NO.',32X,'CONC (MG/M3)',12X,' MAC ')
  130
  131     GO TO 999
  132   900 WRITE(*,*)'IO ERROR IN HDG4= ',IOVAL
  133   999 RETURN
  134     END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0

  135
  136 C    ***************************************************************
  137 C    *     SUBROUTINE HDG5                                         *
  138 C    * PROGRAM TO PRINT HEADING-TOTAL CONT REMOVED BY EACH DEV (MG) *
  139 C    * PRINTS SHEET 1-NO,NAME,CABIN,LEAK,&DEV3..DEV8               *
  140 C    ***************************************************************
  141
  142 C    NOTE:FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
  143
  144     SUBROUTINE HDG5(IDEVNO)
  145
  146 C   INPUTS:
  147 C    IDEVNO=DEVICE NUMBER FOR OUTPUT
  148
  149 C    WRITE HEADING
  150     WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900)
  151   010 FORMAT(1X,24X,'TOTAL CONTAMINANT MASS REMOVED BY EACH DEVICE (MG)
  152    +    SHEET 1')
  153     WRITE(IDEVNO,20,IOSTAT=IOVAL,ERR=900)
  154   020 FORMAT(1X,' NO.',14X,'NAME',16X,'CABIN',9X,'LEAK',8X,'DEV3',
  155    +8X,'DEV4',8X,'DEV5',8X,'DEV6',8X,'DEV7',8X,'DEV8')
```

```
 156
 157     GO TO 999
 158   900 WRITE(*,*)'IO ERROR IN HDG5= ',IOVAL
 159   999 RETURN
 160     END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0

 161
 162 C    ***************************************************************
 163 C    *     SUBROUTINE HDG6                                         *
 164 C    * PROGRAM TO PRINT HEADING-TOTAL CONT REMOVED BY EACH DEV (MG) *
 165 C    * PRINTS SHEET 2-NO,NAME,& DEV9..DEV15                        *
 166 C    ***************************************************************
 167
 168 C    NOTE:FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
 169
 170     SUBROUTINE HDG6(IDEVNO)
 171
 172 C   INPUTS:
 173 C    IDEVNO=DEVICE NUMBER FOR OUTPUT
 174
 175 C    WRITE HEADING
 176     WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900)
 177   010 FORMAT(1X,24X,'TOTAL CONTAMINANT MASS REMOVED BY EACH DEVICE (MG)
 178    +    SHEET 2')
 179     WRITE(IDEVNO,20,IOSTAT=IOERR,ERR=900)
 180   020 FORMAT(1X,' NO.',14X,'NAME',16X,' DEV9',8X,'DEV10',7X,'DEV11',
 181    +7X,'DEV12',7X,'DEV13',7X,'DEV14',7X,'DEV15')
 182
 183     GO TO 999
 184   900 WRITE(*,*)'IO ERROR IN HDG6= ',IOVAL
 185   999 RETURN
 186     END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0

 187
 188 C    ***************************************************************
 189 C    *     SUBROUTINE HDG7                                         *
 190 C    * PROGRAM TO PRINT HEADING-DEVICE REM EFF AT END OF TIME INCR  *
 191 C    * PRINTS NO.,NAME,#2..#12                                     *
 192 C    ***************************************************************
 193
 194 C    NOTE:FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
 195
 196     SUBROUTINE HDG7(IDEVNO)
 197
 198 C   INPUTS:
 199 C    IDEVNO=DEVICE NUMBER FOR OUTPUT
 200
 201 C    WRITE HEADING
 202     WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900)
 203   010 FORMAT(1X,24X,'DEVICE REMOVAL EFFICIENCY AT END OF TIME INCREMENT
 204    + (DEC)')
 205     WRITE(IDEVNO,20,IOSTAT=IOERR,ERR=900)
 206   020 FORMAT(1X,' NO.',14X,'NAME',15X,'#2',4X,'#3',
 207    +4X,'#4',4X,'#5',4X,'#6',4X,'#7',4X,'#8',4X,'#9',3X,'#10',
```

```
  208    +3X,'#11',3X,'#12',3X,'#13',3X,'#14',3X,'#15')
  209
  210     GO TO 999
  211   900 WRITE(*,*)'IO ERROR IN HDG7= ',IOVAL
  212   999 RETURN
  213     END

 NUMBER OF WARNINGS IN PROGRAM UNIT: 0
 NUMBER OF ERRORS  IN PROGRAM UNIT: 0

  214
  215 C   ***************************************************************
  216 C   *     SUBROUTINE HDG8                                         *
  217 C   * PROGRAM TO PRINT HEADING-RATE OF CONT REMOVAL-EACH DEV (MG)  *
  218 C   * PRINTS SHEET 1-NO,NAME,CABIN,LEAK,&DEV3..DEV8               *
  219 C   ***************************************************************
  220
  221 C   NOTE:FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
  222
  223     SUBROUTINE HDG8(IDEVNO)
  224
  225 C   INPUTS:
  226 C    IDEVNO=DEVICE NUMBER FOR OUTPUT
  227
  228 C    WRITE HEADING
  229     WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900)
  230   010 FORMAT(1X,24X,'RATE OF CONTAMINANT REMOVAL-EACH DEVICE (MG/HR)
  231    +   SHEET 1')
  232     WRITE(IDEVNO,20,IOSTAT=IOVAL,ERR=900)
  233   020 FORMAT(1X,' NO.',14X,'NAME',16X,'CABIN',9X,'LEAK',8X,'DEV3',
  234    +8X,'DEV4',8X,'DEV5',8X,'DEV6',8X,'DEV7',8X,'DEV8')
  235
  236     GO TO 999
  237   900 WRITE(*,*)'IO ERROR IN HDG8= ',IOVAL
  238   999 RETURN
  239     END

 NUMBER OF WARNINGS IN PROGRAM UNIT: 0
 NUMBER OF ERRORS  IN PROGRAM UNIT: 0

  240
  241 C   ***************************************************************
  242 C   *     SUBROUTINE HDG9                                         *
  243 C   * PROGRAM TO PRINT HEADING-RATE OF CONT REMOVAL-EACH DEV(MG/HR)*
  244 C   * PRINTS SHEET 2-NO,NAME,& DEV9..DEV15                        *
  245 C   ***************************************************************
  246
  247 C   NOTE:FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
  248
  249     SUBROUTINE HDG9(IDEVNO)
  250
  251 C   INPUTS:
  252 C    IDEVNO=DEVICE NUMBER FOR OUTPUT
  253
  254 C    WRITE HEADING
  255     WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900)
  256   010 FORMAT(1X,24X,'RATE OF CONTAMINANT REMOVAL-EACH DEVICE (MG/HR)
  257    +   SHEET 2')
  258     WRITE(IDEVNO,20,IOSTAT=IOERR,ERR=900)
  259   020 FORMAT(1X,' NO.',14X,'NAME',16X,' DEV9',8X,'DEV10',7X,'DEV11',
```

```
260    +7X,'DEV12',7X,'DEV13',7X,'DEV14',7X,'DEV15')
261
262     GO TO 999
263   900 WRITE(*,*)'IO ERROR IN HDG9= ',IOVAL
264   999 RETURN
265     END
266

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS  IN COMPILATION : 0

RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\LDIGEN. Options: /C 80 /L /BY   05/21/92 12:56:50
    1 C       ************************************************************************
    2 C       *                                                                      *
    3 C       *     SUBROUTINE LDIGEN                                                *
    4 C       *  SUBROUTINE TO LOAD INTERNAL GENERATION FROM MAT CDI COL 1 &         *
    5 C       *  COL 10-22 INTO MAT DD COL 19                                        *
    6         ************************************************************************
    7         SUBROUTINE LDIGEN(I,DD,NROW,NCOL,CDI,NROW2,NCOL2,LIN2)
    8         INTEGER NROW,NCOL,NROW2,NCOL2,LIN2
    9         REAL DD(NROW,NCOL)
   10         REAL CDI(NROW2,NCOL2)
   11
   12 C INPUTS:
   13 C      I=CONTAMINANT NUMBER
   14 C      DD,NROW,NCOL=NAME AND DIMENSIONS OF MAT DD
   15 C      CDI,NROW2,NCOL2=NAME AND DIMENSIONS OF MAT CDI
   16 C      LIN2=NUMBER OF DEVICES IN MAT DD
   17 C OUTPUT
   18 C      LOADS INTERNAL GENERATION FROM MAT CDI INTO MAT DD COL 19
   19 C
   20          DD(1,19)=CDI(I,1)
   21          DD(2,19)=0
   22          DO 10 J=3,LIN2
   23           DD(J,19)=CDI(I,J+7)
   24    010 CONTINUE
   25          RETURN
   26 C        ************ END OF SUBROUTINE LDIGEN ***************************
   27          END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS  IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\LODEFF. Options: /C 80 /L /BY   05/21/92 12:57:05
   1 C      ************************************************************************
   2 C      *       SUBROUTINE LODEFF                                              *
   3 C      * SUBROUTINE TO LOAD LAST INCR EFF FROM MAT CC INTO MAT DD COL 20*
   4 C      *   USES ADJUSTABLE SIZE ARRAYS                                        *
   5 C      ************************************************************************
   6        SUBROUTINE LODEFF(I,DD,NROW,NCOL,CC,NROW1,NCOL1,LIN2)
   7        INTEGER NROW,NCOL,NROW1,NCOL1
   8        REAL DD(NROW,NCOL)
   9        REAL CC(NROW1,NCOL1)
  10
  11 C    INPUTS:
  12 C      I=CONTAMINANT LINE NUMBER IN MAT CC
  13 C      DD,NROW,NCOL=NAME & DIMENSIONS OF MAT DD
  14 C      CC,NROW1,NCOL1=NAME & DIMENSIONS OF MAT CC
  15 C      LIN2=NO. OF DEVICES IN MAT DD
  16
  17        DD(1,20)=0
  18        K=7
  19        DO 100 J=2,LIN2
  20         DD(J,20)=CC(I,K)
  21         K=K+3
  22  100 CONTINUE
  23        RETURN
  24 C      ************* END OF SUBROUTINE LODEFF ************************
  25        END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS  IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\MASBAL. Options: /C 80 /L /BY   05/21/92 12:57:51
     1 C     ************************************************************
     2 C     *       MASS BALANCE SUBROUTINE-MASBAL                     *
     3 C     * FOR 1 CONT AT A TIME AT A GIVEN DEVICE EFFICIENCY CALCULATES *
     4 C     * CAV,CFINAL,CEQ,M.REMOVED (ALL DEV+CABIN)-DATA PUT IN MAT DD  *
     5 C     ************************************************************
     6 C NOTE: BEFORE RUNNING THIS SUBROUTINE MUST ZERO MAT DD COL 17-21
     7 C      (DONE BY PRAFIL) & LOAD REM EFF FOR EACH DEVICE INTO
     8 C      MAT DD COL 20 (DONE BY LODEFF OR CNRSUB)
     9
    10       SUBROUTINE MASBAL(I,TN,TN1,DD,NROW,NCOL,CC,NROW1,NCOL1,
    11      +CAVCLC,CDI,NROW2,NCOL2,CFINAL,CEQLIB,LIN,LIN2)
    12       INTEGER NROW,NCOL,NROW1,NCOL1,NROW2,NCOL2
    13       REAL DD(NROW,NCOL)
    14       REAL CC(NROW1,NCOL1)
    15       REAL CDI(NROW2,NCOL2)
    16 C
    17 C SUBROUTINES REQUIRED:
    18 C CALCM-CALCULATE CIN,COUT,M.REM,SUM MASS REM-IN MAT DD
    19 C LDIGEN-LOAD INTERNAL M.GEN FOR DEVICE + CABIN FROM CDI INTO DD COL 19
    20 C PCAVCF-USING CEQ & CINIT, CALC CFINAL & CAVERAGE
    21
    22 C INPUTS:
    23 C FROM PCSET, PREDCT, AND CONVRG
    24 C   I=CONTAMINANT NO.
    25 C   TN=INCREMENT END TIME(HRS); TNI=INCR BEGINNING TIME HRS
    26 C   DD,NROW,NCOL=NAME AND SIZE OF MAT DD
    27 C   CC,NROW1,NCOL1=NAME AND SIZE OF MAT CC
    28 C   CDI,NROW2,NCOL2=NAME AND SIZE OF MAT CDI
    29 C   LIN=NO. OF CONT IN MAT CDI
    30 C   LIN2=NO. OF DEVICES IN MAT DD
    31 C FROM CALCM
    32 C   SMREM=SUM OF MASS REM FOR ALL DEVICES (MG/HR)-TOTAL OF DD COL 21
    33 C   SMGEN=SUM OF MASS GEN IN ALL DEVICES INCL CABIN(MG/HR)-DD COL 19
    34 C FROM LDIGEN
    35 C   IT LOADS CABIN M.GEN (MG/HR) FROM MAT CDI INTO DD(1,19)
    36 C   IT LOADS M.GEN DEVICES FROM MAT CDI COL 2-15,19 INTO DD COL 19
    37 C FROM PCAVCF
    38 C   CAVCLC=CALC INCR CABIN CONT CONC (MG/CU M)
    39 C   CFINAL=FINAL INCR CABIN CONT CONC (MG/CU M)
    40 C OUTPUTS:
    41 C TO PCSET, PREDCT, AND CONVRG
    42 C   CAVCLC=CALCULATED CABIN AVERAGE CONC(MG/CU M)
    43 C   CEQLIB=CABIN EQUILIBRIUM CONCENTRATION (MG/CU M)
    44 C   CFINAL=INCREMENT FINAL CABIN CONCENTRATION (MG/CU M)
    45 C   PUTS M.REM FOR CABIN + DEVICES IN MAT DD COL 21
    46 C TO CALCM
    47 C   CAV=CABIN CONT CONC (MG/CU M)
    48 C   DD(1,19)=50 (CABIN M.GEN)
    49 C   OTHER DEVICES DD(2-15,19) MUST =0 AT THIS POINT (SEE PRAFIL)
    50 C TO LDIGEN
    51 C    I=CONTAMINANT NUMBER
    52 C    DD,NROW,NCOL=NAME AND DIMENSIONS OF MAT DD
    53 C    CDI,NROW2,NCOL2=NAME AND DIMENSIONS OF MAT CDI
    54 C    LIN2=NUMBER OF DEVICES IN MAT DD
    55 C TO PCAVCF
    56 C   TN,TN1
    57 C   CINIT=INITIAL INCR CABIN CONT CONC (MG/CU M)=CC(I,1)
    58 C   CEQLIB=CABIN EQULIB CONC (MG/CU M)
```

```
C   SQEFFN=SUM OF Q*REM EFF NET FOR ALL DEVICES (CU M/HR)
C   CVOL=CABIN VOL (CU M)=DD(1,9)
C   SMNTC=SUM OF MASS NET TO CABIN(MG/HR)

C CABIN VOL (CU M)
      CVOL=DD(1,9)
C CINITIAL (MG/CU M)
      CINIT=CC(I,1)

C  EVALUATE SUM Q*REM EFF NET USING M.GEN IN DEVICES=0 (DD COL 19)
C   SET CABIN AVERAGE CONCENTRATION = TO ARBITRARY VALUE OF 100
C   AND INTERNAL GENERATION IN DEVICES =0 (NOT YET LOADED)
C      SET CABIN M.GEN=ARBITRARY VALUE OF 50 (DD(1,19))
C      SMGEN=SUM M.GEN IN ALL DEVICES +CABIN (MG/HR)
C      SMREM=SUM M. REMOVED BY ALL DEVICES (MG/HR)
C      CAV=CABIN AVERAGE CONCENTRATION (MG/CU M)
      CAV=100
      DD(1,19)=50

      CALL CALCM(DD,NROW,NCOL,CAV,SMGEN,SMREM,TN,LIN2)
C     SQEFFN=SUM OF Q*REMOVAL EFF NET (MG/HR)
      SQEFFN=SMREM/CAV

C
C LOAD INTERNAL GENERATION FOR ALL DEV+CABIN FROM CDI INTO DD COL 19
      CALL LDIGEN(I,DD,NROW,NCOL,CDI,NROW2,NCOL2,LIN2)
C
C EVALUATE SUM OF M.NET TO CABIN=M.GEN CABIN+SUM M.GEN ALL DEVICES -
C     SUM M.REM ALL DEVICES
C     SMNTC=SUM M.NET TO CABIN=AMT GEN WHICH GETS TO THE CABIN DIRECTLY
C     SET C CABIN AV=0
      CAV=0
C     GET SUM MASS GEN CABIN+ INTERNAL DEVICES AND SUM MASS REMOVED ALL
C     DEVICES FROM SUBROUTINE-SINCE CABIN C=0 NO CABIN CONT WILL BE REM
      CALL CALCM(DD,NROW,NCOL,CAV,SMGEN,SMREM,TN,LIN2)
      SMNTC=SMGEN-SMREM
C      NOTE:SMNTC IS ALSO PUT IN DD(1,21) BY CALCM
C
C GET CALCULATED CABIN EQUILIBRIUM CONCENTRATION (CAVCLC) (MG/CU M)

C
      IF(SQEFFN.LT.1E-10) THEN
       IF(CVOL.EQ.0) THEN
        CFINAL=1E10
       ELSE
        CFINAL=CINIT+(TN-TN1)*SMGEN/CVOL
       ENDIF
       CAVCLC=(CINIT+CFINAL)/2
       CEQLIB=1E10
      ELSE
       CEQLIB=SMNTC/SQEFFN
C      CALCULATE CAVCLC AND CFINAL FROM SUBROUTINE
       CALL PCAVCF(TN,TN1,CEQLIB,SQEFFN,CVOL,SMNTC,CINIT,
     + CAVCLC,CFINAL)
      ENDIF
C USING CAV CALC EVALUATE M.REM FOR CABIN + DEVICES AND PUT
C     IN MAT DD COL 21
      CAV=CAVCLC
      CALL CALCM(DD,NROW,NCOL,CAV,SMGEN,SMREM,TN,LIN2)
C
```

```
  119        RETURN
  120 C END OF SUBROUTINE MASBAL
  121 C ************** END OF SUBROUTINE MASBAL ****************************
  122        END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS  IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\FORTRAN\TCC\MCALC. Options: /C 80 /L /BIJY  03/15/94 10:40:19

  1 C   ***************************************************************
  2 C   *         SUBROUTINE MCALC                                      *
  3 C   *  MAIN CALCULATION LOOP SUBROUTINE FOR 1 TIME INCREMENT        *
  4 C   *  FOR ALL CONTAMINANTS ONE AT A TIME                           *
  5 C   *  BASED ON SUM MASS REM LAST INCR, FOR EACH CONT               *
  6 C   *  CALCULATE NEW REMOVAL EFF, CAV CALC CABIN,                   *
  7 C   *  CEQUILIB, CFINAL, & M.REMOVED ALL DEVICES-PUT IN MAT CC      *
  8 C   ***************************************************************
  9     SUBROUTINE MCALC(I,TN,TN1,DD,NROW,NCOL,
 10    +CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,LIN,LIN2,
 11    +PRTSW2,PRTSW3,PRTSW4,IMSGDN)
 12     INTEGER NROW,NCOL,NROW1,NCOL1,NROW2,NCOL2,PRTSW2,PRTSW3,PRTSW4,KK
 13     CHARACTER NN(NROW1)*30
 14     REAL DD(NROW,NCOL)
 15     REAL CC(NROW1,NCOL1)
 16     REAL CDI(NROW2,NCOL2)
 17 C SUBROUTINES REQUIRED:
 18 C  PREDCT=PREDICT CAV BASED ON M.GEN OF THIS INCR & REM EFF OF LST INC
 19 C  CONVRG=CALC CAV CALC,CEQ,CFINAL,M.REM,REM EFF
 20 C  CROUT=PRINT TEST VALUES OF MAT CC
 21 C  RROUT=PRINT TEST VALUES OF MAT DD
 22 C
 23 C INPUTS:
 24 C FROM MAIN PROG
 25 C  I=CONTAMINANT NO.
 26 C  TN,TN1 =INCREMENT END & BEGINNING TIME (HRS)
 27 C  DD,NROW,NCOL=NAME & DIM OF MAT DD
 28 C  CC,NROW1,NCOL1=NAME & DIM OF MAT CC
 29 C  CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
 30 C  NN=NAME OF MAT NN
 31 C  LIN=NUMBER OF CONTAMINANTS IN MAT NN & CDI
 32 C  LIN2=NO. DEVICES IN MAT DD
 33 C FROM PREDCT
 34 C  CAVPRD=PRED CABIN CONT CONC (MG/CU M)
 35 C FROM CONVRG
 36 C  CAVCLC=CALC CABIN CONT CONC (MG/CU M)
 37 C  CFINAL=FINAL INCR CABIN CONT CONC(MG/CUM)
 38 C  KK=COUNTER FOR CONVERGENCE
 39 C  CEQLIB=EQUILIBRIUM CABIN CONT CONC(MG/CU M)
 40 C  IMSGDN=DEVICE NO FOR MESSAGE AND TEXT PRINTOUT OUTPUT
 41 C OUTPUTS:
 42 C TO PREDCT
 43 C  I=CONTAMINANT NO.
 44 C  TN,TN1 =INCREMENT END & BEGINNING TIME (HRS)
 45 C  DD,NROW,NCOL=NAME & DIM OF MAT DD
 46 C  CC,NROW1,NCOL1=NAME & DIM OF MAT CC
 47 C  CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
 48 C  LIN=NO. OF CONTAMINANTS IN MAT CDI
 49 C  LIN2=NO. DEVICES IN MAT DD
 50 C TO CONVRG
 51 C  I=CONTAMINANT NO.
 52 C  TN,TN1 =INCREMENT END & BEGINNING TIME (HRS)
 53 C  DD,NROW,NCOL=NAME & DIM OF MAT DD
 54 C  CC,NROW1,NCOL1=NAME & DIM OF MAT CC
 55 C  CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
 56 C  LIN=NO. OF CONTAMINANTS IN MAT CDI
 57 C  LIN2=NO. DEVICES IN MAT DD
```

```
C  CAVPRD=PREDICTED CABIN CONC FOR INCREMENT (MG/CU M)
C TO MAT CC
C  PUTS CAVCLC,CEQLIV,&CFINAL IN CC(I,2-3 &4)
C  PUTS REM EFF FROM DD COL20 IN CC(I,7-10-13 ETC)
C  PUTS M.REM FOR EACH DEV FROM DD COL21 IN CC(I,6-9-12 ETC)
C  PUTS SUM MASS REM FOR EACH DEV IN CC(I,8-11-14 ETC)
C
C
C  BEGIN LOOP FOR EACH CONTAMINANT FOR EACH TIME INCREMENT
   DO 100 I=1,LIN
C   CALC CAV PRED CABIN FOR CONT BASED ON REM EFF OF LAST INCREMENT
C   AND GENERATION RATES OF THIS INCREMENT
   CALL PREDCT(I,TN,TN1,CAVPRD,DD,NROW,NCOL,
  +CC,NROW1,NCOL1,CDI,NROW2,NCOL2,CAVCLC,CFINAL,CEQLIB,LIN,LIN2,NN)

C
C   CONVERGE UNTIL CCALC=CPRED
   CALL CONVRG(I,TN,TN1,CAVPRD,DD,NROW,NCOL,
  +CC,NROW1,NCOL1,CDI,NROW2,NCOL2,CAVCLC,CFINAL,CEQLIB,KK,LIN,
  +LIN2,NN,PRTSW3,IMSGDN)
C
C   IF KK>20 THEN BEGIN 1/20 TIME INCREMENT CONVERGENCE ROUTINE
    IF (KK.GT.20) THEN
     KK=1
C     BEGIN 1/20 INCREMENT CONVERGENCE ROUTINE
C     NEW INCREMENT INITIAL TIME (HRS)
      TN1NEW=TN1
C     NEW TIME INCREMENT (HRS)
      BINEW=(TN-TN1)/20
C     BEGIN LOOP FOR 1/20 INCREMENT SIZE TIME INCREMENT
C      NEW INCREMENT FINAL TIME (HRS)
 200   TNNEW=TN1NEW+BINEW

C     ZERO MAT DD COL 17-21
      CALL PRAFIL(DD,NROW,NCOL,17,21)
C     LOAD EFFICIENCY FROM LAST INCREMENT INTO MAT DD COL 20
      CALL LODEFF(I,DD,NROW,NCOL,CC,NROW1,NCOL1,LIN2)
C     CALC CAV PRED CABIN FOR CONT BASED ON REM EFF OF LAST INCREMENT
C     AND GENERATION RATES OF THIS INCREMENT
     CALL PREDCT(I,TNNEW,TN1NEW,CAVPRD,DD,NROW,NCOL,
   +  CC,NROW1,NCOL1,CDI,NROW2,NCOL2,CAVCLC,CFINAL,CEQLIB,LIN,
   +  LIN2,NN)
C
C     CONVERGE UNTIL CCALC=CPRED
     CALL CONVRG(I,TNNEW,TN1NEW,CAVPRD,DD,NROW,NCOL,
   +  CC,NROW1,NCOL1,CDI,NROW2,NCOL2,CAVCLC,CFINAL,CEQLIB,KK,LIN,
   +  LIN2,NN,PRTSW3,IMSGDN)
C
C     IF KK>20 THEN PRINT CONVERGENCE WARNING
     IF (KK.GT.20) THEN
      OPEN (IMSGDN,FILE='CON',IOSTAT=IOVAL)
      WRITE(IMSGDN,*)'WARNING: CALCULATION DID NOT CONVERGE FOR'
      WRITE(IMSGDN,*)'      FULL AND 1/20 INCREMENT ROUTINES'
      WRITE(IMSGDN,50) I,NN(I),TN1NEW,TNNEW
 050    FORMAT (1X,'CONT NO.= ',I4,2X,A,/,1X,
   +    'FOR INCREMENT INIT & FINAL TIMES= ',F8.2,F8.2)
      CLOSE(IMSGDN)
     ENDIF

C     FILL MAT CC WITH RESULTS
```

```
118 C     PUT CAVCLC,CEQLIB, AND CFINAL IN CC
119       CC(I,2)=CAVCLC
120       CC(I,3)=CEQLIB
121       CC(I,4)=CFINAL
122 C     PUT REM EFF FROM LAST ITER DD COL 20 IN CC(I,7-10-13ETC)
123       K=7
124       DO 102 J=2,LIN2
125        CC(I,K)=DD(J,20)
126        K=K+3
127  102   CONTINUE
128
129 C     TAKE CABIN M.REM(MG/HR) FROM DD(1,21) & PUT IN MAT CC(I,5)
130       CC(I,5)=DD(1,21)
131 C     TAKE M.REM FROM DD COL 21 & PUT IN CC(I,8-11-14ETC)
132       K=8
133       DO 103 J=2,LIN2
134        CC(I,K)=DD(J,21)
135        K=K+3
136  103   CONTINUE
137
138 C     CALCULATE SUM OF MASS REMOVED IN DEVICES + CABIN TO DATE AND
139 C     PUT IN CC(I,6-9-12ETC)
140      K=5
141      DO 104 J=1,LIN2
142       CC(I,K+1)=CC(I,K+1)+CC(I,K)*(TNNEW-TN1NEW)
143       K=K+3
144  104  CONTINUE
145 C
146 C     IF PRTSW2=1 THEN PRINT MAT CC INFO FOR THIS CONTAMINANT
147       IF (PRTSW2.EQ.1) THEN
148        OPEN (IMSGDN,FILE='CON',IOSTAT=IOVAL)
149         WRITE(IMSGDN,*)'NINC,TN,TN1 ',NINC,TN,TN1
150         WRITE(IMSGDN,*)'PRINTOUT FOR ONE CONT INSIDE 1/20 INCR
151    + LOOP OF MCALC'
152          WRITE(IMSGDN,*)'INFO FROM MAT CC'
153        CLOSE(IMSGDN)
154        CALL CROUT(NN,CC,NROW1,NCOL1,1,NCOL1,LIN,I,I,IMSGDN)
155      ENDIF
156 C
157
158 C     REPEAT LOOP FOR 1/10 INCREMENT IF END OF 1/20 INCREMENT TIME
159 C     (TNNEW) IS < THAN END OF LARGER TIME INCR (TN)
160 C     ELSE IF TNNEW>=TN, END 1/20 ENCR CONV & PRINT ANSWERS+REPEAT
161 C      FOR ANOTHER CONTAMINANT
162
163      IF (TNNEW.LT.TN) THEN
164 C      RESET FOR ANOTHER 1/20 TIME INCREMENT
165       TN1NEW=TNNEW
166       CC(I,1)=CC(I,4)
167       GO TO 200
168      ELSE
169 C      END 1/20 INCR CONV ROUTINE-REPEAT FOR ANOTHER CONT
170       GOTO 100
171      ENDIF
172
173 C    END OF CONVERGENCE ROUTINE
174     ENDIF
175
176 C    CALC SUM MASS REMOVED & FILL MAT CC WITH RESULTS
177 C    PUT CAVCLC,CEQLIB, AND CFINAL IN CC
```

```
178       CC(I,2)=CAVCLC
179       CC(I,3)=CEQLIB
180       CC(I,4)=CFINAL
181  C    PUT REM EFF FROM LAST ITER DD COL 20 IN CC(I,7-10-13ETC)
182       K=7
183       DO 302 J=2,LIN2
184        CC(I,K)=DD(J,20)
185        K=K+3
186   302  CONTINUE
187
188  C    TAKE CABIN M.REM(MG/HR) FROM DD(1,21) & PUT IN MAT CC(I,5)
189      CC(I,5)=DD(1,21)
190  C    TAKE M.REM FROM DD COL 21 & PUT IN CC(I,8-11-14ETC)
191       K=8
192       DO 303 J=2,LIN2
193        CC(I,K)=DD(J,21)
194        K=K+3
195   303  CONTINUE
196
197  C    CALCULATE SUM OF MASS REMOVED IN DEVICES + CABIN TO DATE AND PUT
198  C    IN CC(I,6-9-12ETC)
199      K=5
200      DO 304 J=1,LIN2
201       CC(I,K+1)=CC(I,K+1)+CC(I,K)*(TN-TN1)
202       K=K+3
203   304 CONTINUE
204  C
205
206  C    IF PRTSW4=1 THEN PRINT MAT DD+MAT CC INFO FOR THIS CONTAMINANT
207       IF (PRTSW4.EQ.1) THEN
208        OPEN (IMSGDN,FILE='CON',IOSTAT=IOVAL)
209         WRITE(IMSGDN,*)'PRINTOUT FOR ONE CONT AT END OF MCALC'
210         WRITE(IMSGDN,*)'INFO FROM MAT CC'
211        CLOSE(IMSGDN)
212        CALL CROUT(NN,CC,NROW1,NCOL1,1,NCOL1,LIN,I,I,IMSGDN)
213        OPEN (IMSGDN,FILE='CON',IOSTAT=IOVAL)
214         WRITE(IMSGDN,*)'INFO FROM MAT DD'
215        CLOSE(IMSGDN)
216        CALL RROUT(DD,NROW,NCOL,1,NCOL,LIN2,IMSGDN)
217       ENDIF
218  C
219  C    END OF I LOOP FOR EACH CONTAMINANT
220  C
221   100 CONTINUE
222      RETURN
223  C    ****** END OF SUBROUTINE MCALC ************************************
224      END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS  IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\PCAVCF. Options: /C 80 /L /BY   05/21/92 12:57:51
C      ************************************************************************
C      *         SUBROUTINE PCAVCF                                            *
C      * SUBROUTINE TO PREDICT INCREMENT CALCULATED AVERAGE EFF               *
C      * (C AV CALC), FINAL EFF (CFINAL) & CABIN CONTAMINANT                  *
C      * CONCENTRATION                                                        *
C      ************************************************************************

       SUBROUTINE PCAVCF(TN,TN1,SCEQLIB,SQEFFN,CVOL,SMNTC,CINIT,
     + CAVCLC,SCFINAL)
       DOUBLE PRECISION EXPON,CEQLIB,CFINAL
       CEQLIB=DBLE(SCEQLIB)

C SUBROUTINES REQUIRED: NONE
C
C INPUTS:
C TN, TN1=INITIAL & FINAL INCREMENT TIME (HRS)
C SCEQLIB(CEQLIB)=EQUILIBRIUM CABIN CONC (MG/CUM)
C SQEFFN=SUM Q*REMOVAL EFF NET (MG/HR)
C CVOL=CABIN VOLUME (CU M)
C SMNTC=SUM MASS CONT NET TO CABIN (MG/HR)
C CINIT=INITIAL INCREMENT CONT CONC (MG/CU M)
C OUTPUTS:
C SCAVCLC(CAVCLC)=CALC AVERAGE CABIN CONC (MG/CU M)
C SCFINAL(CFINAL)=FINAL INCREMENT CONC (MG/CU M)
C
       IF(CVOL.LE.0) THEN
         CAVCLC=CEQLIB
         CFINAL=CEQLIB
         GOTO 99
       ENDIF
C      CALCULATION FOR CFINAL
       EXPON=(TN-TN1)*SQEFFN/CVOL
       IF(ABS(EXPON).GT.50) THEN
         CAVCLC=CEQLIB
         CFINAL=CEQLIB
         GOTO 99
       ENDIF
       IF(ABS(EXPON).LT.1E-6) THEN
         CFINAL=CINIT+SMNTC*(TN-TN1)/CVOL
         CAVCLC=(CINIT+CFINAL)/2
         CEQLIB=1E10
         GOTO 99
       ELSE
         CFINAL=CINIT+(SMNTC/SQEFFN-CINIT)*(1-EXP(-EXPON))
       ENDIF
C      CALCULATION FOR C AVERAGE CALC
       IF ((CINIT.EQ.CFINAL).OR.(CFINAL.EQ.CEQLIB)) THEN
        CAVCLC=CFINAL
        GOTO 99
       ENDIF
       IF ((CEQLIB-CINIT)/(CEQLIB-CFINAL).LT.1E-6) THEN
         CAVCLC=(CINIT+CFINAL)/2
       ELSE
         CAVCLC=CEQLIB-(CFINAL-CINIT)/LOG((CEQLIB-CINIT)/
     + (CEQLIB -CFINAL))
       ENDIF
099 CONTINUE
       SCFINAL=REAL(CFINAL)
```

```
   59          RETURN
   60 C        ********** END OF SUBROUTINE PCAVCF ***************************
   61          END
   62 C

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS  IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\PCSET.F Options: /C 80 /L /BY   05/21/92 12:58:31
   1 C      ************************************************************
   2 C      *         SUBROUTINE  PCSET                                *
   3 C      *    SUBROUTINE FOR PRECALCULATION SETUP ROUTINE           *
   4 C      *    FOR ALL CONTAMINANTS ONE AT A TIME                    *
   5 C      *    CALL EFF SUBROUTINES FOR DEVICES; GET CAV CABIN PRED, *
   6 C      *    CEQUILIB, CFINAL, & M.REMOVED ALL DEVICES-PUT IN MAT CC *
   7 C      ************************************************************
   8         SUBROUTINE PCSET(TN1,LIN,DD,NROW,NCOL,CC,NROW1,NCOL1,
   9        +CDI,NROW2,NCOL2,LIN2,NN,PRTSW1,IMSGDN)
  10         INTEGER NROW,NCOL,NROW1,NCOL1,NROW2,NCOL2,PRTSW1,KK
  11         CHARACTER NN(NROW1)*30
  12         REAL DD(NROW,NCOL)
  13         REAL CC(NROW1,NCOL1)
  14         REAL CDI(NROW2,NCOL2)
  15         KK=0
  16 C SUBROUTINES REQUIRED:
  17 C    PRAFIL-ZERO MAT DD COL 17-21
  18 C    CNRSUB-USING CAV=1E-20, FIND REMOVAL EFF AND PUT IN MAT DD COL 20
  19 C    MASBAL-CALC CAV CALC PRED, CEQ, CFINAL, M.REM
  20 C    CROUT-TEST PRINTOUT OF CONT INFO
  21 C
  22 C INPUTS:
  23 C FROM MAIN CALC LOOP
  24 C    TN1=INCREMENT INITIAL TIME (HRS)
  25 C    LIN=NO. OF CONT IN MAT CC AND NN
  26 C    DD,NROW,NCOL=NAME & SIZE OF MAT DD
  27 C    CC,NROW1,NCOL1=NAME & SIZE OF MAT CC
  28 C    CDI,NROW2,NCOL2=NAME & SIZE OF MAT CDI
  29 C    LIN2=NO. DEVICES IN MAT DD
  30 C    NN=NAME OF MAT NN
  31 C    PRTSW1=PRINTSWITCH WHICH CONTROLS TEST PRINTOUT
  32 C    IMSGDN=DEVICE NUMBER FOR MESSAGES AND TEST PRINTOUT
  33 C FROM PRAFIL
  34 C    PUTS ZEROS IN MAT DD COL 17-21
  35 C FROM CNRSUB
  36 C    CNRSUB PUTS REM EFF(DEC) FOR EACH DEVICE IN MAT DD COL 20
  37 C FROM MASBAL (PREDICTED VALUES)
  38 C    CAVCLC=AVERAGE CABIN CONC (MG/CU M)
  39 C    CFINAL=FINAL INCREMENT CONT CONC (MG/CU M)
  40 C    CEQLIB=EQUILIBRIUM CONT CONC (MG/CU M)
  41 C    M.REM FOR ALL DEVICES PLACED IN COL 21 OF MAT DD
  42 C OUTPUTS
  43 C TO MAIN PROGRAM
  44 C    PUT IN MAT CC
  45 C      CAVPRD=PRED CABIN AV CONC (MG/CU M): =CC(I,2)
  46 C      CEQLIB=EQUILIBRIUM CABIN CONT CONC (MG/CU M):=CC(I,3)
  47 C      CFINAL=FINAL CABIN CONT CONC (MG/CU M):=CC(I,4)
  48 C      PUTS REM EFF FROM DD COL 20 IN CC(I,7-10-13-16 ETC)
  49 C      PUTS M.REM IN CC(I,5-8-11-14...)
  50 C TO PRAFIL
  51 C     NAME AND SIZE OF MAT DD+FIRST & LAST COLUMN TO ZERO
  52 C TO CNRSUB
  53 C    I=CONT NO.
  54 C    TN,TN1=CONT INCREMENT FINAL,INIT TIME (HRS)
  55 C    CAVPRD=CABIN AVERAGE CONT CONC (MG/CU M)
  56 C    DD,NROW,NCOL=NAME & DIMENSIONS OF MAT DD
  57 C    CC,NROW1,NCOL1=NAME & DIM OF MAT CC
  58 C    CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
```

```
C    LIN2=NO. ACTIVE DEVICES IN MAT DD
C TO MASBAL
C    I=CONT NO.
C    TN,TN1=CONT INCREMENT FINAL,INIT TIME (HRS)
C    DD,NROW,NCOL=NAME & DIMENSIONS OF MAT DD
C    CALCLC=CALC CABIN AV CONC (MG/CU M)
C    CC,NROW1,NCOL1=NAME & DIM OF MAT CC
C    CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
C    CFINAL=CABIN FINAL CONCENTRATION (MG/CU M)
C  CEQLIB=CABIN EQUILIBRIUM CONCENTRATION (MG/CU M)
C  LIN=NO. OF CONTAMINANTS IN MAT CDI
C  LIN2=NO. ACTIVE DEVICES IN MAT DD
C TO RROUT
C  MATRIX NAME,#ROWS,#COLS,FIRST & LAST COL TO PRINT,#LINES TO PRINT

C  BASIC TIME INCREMENT (HRS)
       BINC=DD(1,11)
C  SET FINAL INCREMENT TIME (HRS)
       TN=0.1*BINC/24
C  BEGIN LOOP FOR EACH CONTAMINANT - ONE AT A TIME
C      CALCULATE REM EFF FOR EACH DEVICE, GET M.REM, CAV CABIN CALC
C      CEQULIB, CFINAL-PUT IN MAT CC
       DO 100 I=1,LIN
C        ZERO MAT DD COL 17 TO 21
           CALL PRAFIL(DD,NROW,NCOL,17,21)
C        SET CAVPRD = MINIMUM VALUE TO ALLOW COMPUTATION
           CAVPRD=1E-20
           DD(J,22)=CAVPRD
C        CALC REMOVAL EFFICIENCYS (THROUGH EFF CALLING SUBROUTINE)
C          THIS STORES REM EFF IN MAT DD COL 20 FOR EACH DEVICE
         CALL CNRSUB(I,TN,TN1,DD,NROW,NCOL,CC,NROW1,NCOL1,
     +   CDI,NROW2,NCOL2,LIN2,KK)
C        CALL MASS BALANCE-GET CAVPRD(=CAVCLC IN MASBAL),CEQ,CFINAL,M.REM
           CALL MASBAL(I,TN,TN1,DD,NROW,NCOL,CC,NROW1,NCOL1,
     +   CAVCLC,CDI,NROW2,NCOL2,CFINAL,CEQLIB,LIN,LIN2)
           CAVPRD=CAVCLC
C        PUT CAVPRD,CEQLIB,& CFINAL IN MAT CC
           CC(I,2)=CAVPRD
           CC(I,3)=CEQLIB
           CC(I,4)=CFINAL
C        GET REM EFF FROM DD COL 20 AND PUT IN CC(I,7-10-13 ETC)
           K=7
           DO 101 J=2,LIN2
             CC(I,K)=DD(J,20)
             K=K+3
 101       CONTINUE
C    TAKE M.REMOVED FROM MAT DD COL 21 AND PUT IN MAT CC(I,5-8-...)
C          CABIN REMOVAL RATE
           CC(I,5)=DD(1,21)
C          DEVICE 2-15 REMOVAL RATE
           K=8
           DO 102 J=2,LIN2
             CC(I,K)=DD(J,21)
             K=K+3
 102       CONTINUE
C    IF PRTSW1=1 THEN PRINT MAT DD+MAT CC INFO FOR THIS CONTAMINANT
           IF (PRTSW1.EQ.1) THEN
             OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
      WRITE(IMSGDN,*)'PRINTOUT FOR ONE CONT AT END OF PCSET'
             WRITE(IMSGDN,*)'INFO FROM MAT CC'
```

```
 119                CLOSE (IMSGDN)
 120                CALL CROUT(NN,CC,NROW1,NCOL1,1,NCOL1,LIN,I,I,IMSGDN)
 121                OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
 122                WRITE(IMSGDN,*)'INFO FROM MAT DD'
 123                CLOSE (IMSGDN)
 124                CALL RROUT(DD,NROW,NCOL,1,NCOL,LIN2,IMSGDN)
 125              ENDIF
 126 C
 127  100 CONTINUE
 128 C
 129         RETURN
 130 C    ********** END OF SUBROUTINE PCSET ****************************
 131         END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS  IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\PRAFIL. Options: /C 80 /L /BY   05/21/92 12:58:44
   1 C      ***********************************************************************
   2 C      *       SUBROUTINE PRAFIL                                             *
   3 C      *  SUBROUTINE TO FILL ADJUSTABLE SIZE REAL ARRAY WITH ZEROS           *
   4 C      *  PARTIAL FILL-FROM COL FSTCOL TO COL LSTCOL                         *
   5 C      ***********************************************************************
   6        SUBROUTINE PRAFIL(X,NROW,NCOL,FSTCOL,LSTCOL)
   7        INTEGER NCOL,NROW,FSTCOL,LSTCOL
   8        REAL X(NROW,NCOL)
   9
  10 C  INPUTS:
  11 C      X,NROW,NCOL=NAME AND DIMENSIONS OF MATRIX X
  12 C      FSTCOL,LSTCOL=FIRST AND LAST COLUMN TO FILL WITH ZEROS
  13
  14        IF(FSTCOL.GT.NCOL) FSTCOL=NCOL
  15        IF(LSTCOL.GT.NCOL) LSTCOL=NCOL
  16        IF(FSTCOL.GT.LSTCOL) FSTCOL=LSTCOL
  17        DO 110 I=1,NROW
  18        DO 100 J=FSTCOL,LSTCOL
  19         X(I,J)=0.0
  20 100 CONTINUE
  21 110 CONTINUE
  22        RETURN
  23 C      *************  END OF SUBROUTINE PRAFIL ************************
  24        END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS  IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\PRFANS. Options: /C 80 /L /BY   05/21/92 12:58:58
    1 C FILE PRFANS
    2 C      ************************************************************************
    3 C      *     SUBROUTINE PRCDA                                                 *
    4 C      * PROGRAM TO PRINT ANSWERS FOR CONCENTRATION DATA                      *
    5 C      ************************************************************************
    6
    7 C      NOTES:(1)FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
    8 C          (2)IDEVNO MUST BE 6 FOR FORM FEEDS TO BE PRINTED
    9
   10         SUBROUTINE PRCDA(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
   11     +IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR,
   12     +PRTSW8,PRTSW9,FCPLOT,IDEVN1)
   13 C   SUBROUTINES REQUIRED:
   14 C       HDG1,HDG2,HDG3,HDG4
   15
   16 C       TN,TN1=FINAL AND INITIAL INCREMENT TIME (HRS)
   17 C       LIN=TOTAL NUMBER OF CONTAMINANTS
   18 C       CC,NROW1,NCOL1=NAME & SIZE OF MAT CC
   19 C       CDI,NROW2,NCOL2=NAME & SIZE OF MAT CDI
   20 C       NN=NAME OF MAT NN
   21 C       IDEVNO=DEVICE NUMBER FOR OUTPUT
   22 C       NINC=TIME INCREMENT NUMBER
   23 C        =0 THEN PRINT HDG3 WITH PCALC
   24 C        =-1 THEN PRINT HDG3 WITH FINAL
   25 C        ELSE PRINT HDG2 WITH INCREMENT NUMBER
   26 C       IMONTH..IMINUTE=TIME AND DATE INFO
   27 C       FNAME=FILE NAME OUTPUT DATA IS STORED ON
   28 C       IPGCTR=COUNTER FOR SEQUENTIAL PAGE NUMBERS ON ALL OUTPUT
   29
   30         REAL CC(NROW1,NCOL1)
   31         REAL CDI(NROW2,NCOL2)
   32         CHARACTER CNAME*30,FNAME*24,ECHR*1,FCPLOT*24
   33         CHARACTER NN(NROW1)*30
   34         INTEGER PRTSW8,PRTSW9,NINC
   35 C       ECHR=EXCEEDS MAC CHARACTER (Y OR N)
   36         IF ((PRTSW8.EQ.1).OR.((PRTSW8.EQ.0).AND.(NINC.EQ.-1))) THEN
   37 C       INCREMENT PAGE COUNTER BY ONE
   38         IPGCTR=IPGCTR+1
   39
   40 C       START FIRST PAGE
   41 C       PRINT FORM FEED
   42          WRITE(IDEVNO,20,IOSTAT=IOVAL,ERR=900)
   43  020   FORMAT('1')
   44 C       PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 &4
   45           WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)
   46  040   FORMAT(1X,'')
   47           CALL HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)
   48           IF(NINC.EQ.0) THEN
   49             CALL HDG3(1,TN1,TN,IDEVNO)
   50           ELSEIF(NINC.EQ.-1) THEN
   51             CALL HDG3(2,TN1,TN,IDEVNO)
   52           ELSE
   53             CALL HDG2(NINC,TN1,TN,IDEVNO)
   54           ENDIF
   55           CALL HDG4(IDEVNO)
   56 C       PRINT ANOTHER BLANK LINE
   57           WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)
   58
```

```
 59 C       BEGIN LOOP FOR EACH CONTAMINANT 1 TO LIN
 60         DO 100 I=1,LIN
 61
 62 C          CNAME=CONTAMINANT NAME
 63            CNAME=NN(I)
 64 C          FCONC=FINAL CONT CONCENTRATION (MG CU M)
 65            FCONC=CC(I,4)
 66 C          RMAC=MAXIMUM ALLOWABLE CONCENTRATIION (MG/CU M)
 67            RMAC=CDI(I,9)
 68
 69 C          IF CABIN CONC>MAC PRINT 'Y' OTHERWISE PRINT 'N'
 70            IF(FCONC.GT.RMAC) THEN
 71              ECHR='Y'
 72            ELSE
 73              ECHR='N'
 74            ENDIF
 75 C
 76 C          PRINT 56 LINES OF DATA AND THEN START NEW PAGE
 77            WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900) I,CNAME,FCONC,RMAC,ECHR
 78  010   FORMAT(1X,I4,1X,A,1X,G11.4,1X,G11.4,5X,A)
 79
 80 C          CHECK FOR 56 LINES-IF SO, INCREMENT PAGE NUMBER+START NEW PAGE
 81            IF(INT(REAL(I)/56).EQ.REAL(I)/56) THEN
 82              IPGCTR=IPGCTR+1
 83 C            START SUBSEQUENT PAGES
 84 C            PRINT FORM FEED
 85               WRITE(IDEVNO,50,IOSTAT=IOVAL,ERR=900)
 86  050          FORMAT('1')
 87 C            PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 &4
 88              WRITE(IDEVNO,30,IOSTAT=IOVAL,ERR=900)
 89  030          FORMAT(1X,'')
 90               CALL
HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)
 91               IF(NINC.EQ.0) THEN
 92                CALL HDG3(1,TN1,TN,IDEVNO)
 93               ELSEIF(NINC.EQ.-1) THEN
 94                CALL HDG3(2,TN1,TN,IDEVNO)
 95               ELSE
 96                CALL HDG2(NINC,TN1,TN,IDEVNO)
 97               ENDIF
 98               CALL HDG4(IDEVNO)
 99 C             PRINT ANOTHER BLANK LINE
100         WRITE(IDEVNO,30,IOSTAT=IOVAL,ERR=900)
101
102            ENDIF
103  100 CONTINUE
104           ENDIF
105 C
106 C     ********* WRITE CONCENTRATION DATA TO A PLOT FILE *************
107           IF (NINC.NE.-1) THEN
108           IF ((PRTSW9.EQ.1).OR.(PRTSW9.EQ.3)) THEN
109            DO 120 I=1,LIN,300
110               IS=I
111               IE=I+299
112               IF (IE.GT.LIN) IE=LIN
113               WRITE (IDEVN1,110,IOSTAT=IOVAL,ERR=900) TN1,TN,
114     +          (CC(J,4),J=IS,IE)
115  110            FORMAT (T2,2(F8.3,1X),300(G11.4,:,1X))
116  120   CONTINUE
117           ENDIF
```

```
 118         ENDIF
 119         GO TO 999
 120   900 WRITE(*,*)'IO ERROR IN PRCDA= ',IOVAL
 121   999 RETURN
 122         END


NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0

 123
 124 C       ************************************************************************
 125 C       *      SUBROUTINE PRREM1                                               *
 126 C       * PROGRAM TO PRINT ANSWERS-RATE OF CONTAMINAMT REMOVAL (MG/HR)  *
 127 C       * SHEET 1                                                        *
 128 C       ************************************************************************
 129
 130 C    NOTES:(1)FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
 131 C             (2)IDEVNO MUST BE 6 FOR FORM FEEDS TO BE PRINTED
 132
 133         SUBROUTINE PRREM1(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
 134    +IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR)
 135
 136 C   SUBROUTINES REQUIRED:
 137 C       HDG1,HDG2,HDG3,HDG8
 138
 139 C    TN,TN1=FINAL AND INITIAL INCREMENT TIME (HRS)
 140 C       LIN=TOTAL NUMBER OF CONTAMINANTS
 141 C       CC,NROW1,NCOL1=NAME & SIZE OF MAT CC
 142 C    CDI,NROW2,NCOL2=NAME & SIZE OF MAT CDI
 143 C       NN=NAME OF MAT NN
 144 C       IDEVNO=DEVICE NUMBER FOR OUTPUT
 145 C       NINC=TIME INCREMENT NUMBER
 146 C        =0 THEN PRINT HDG3 WITH PCALC
 147 C        =-1 THEN PRINT HDG3 WITH FINAL
 148 C        ELSE PRINT HDG2 WITH INCREMENT NUMBER
 149 C       IMONTH..IMINUTE=TIME AND DATE INFO
 150 C       FNAME=FILE NAME OUTPUT DATA IS STORED ON
 151 C    IPGCTR=COUNTER FOR SEQUENTIAL PAGE NUMBERS ON ALL OUTPUT
 152
 153         REAL CC(NROW1,NCOL1)
 154         REAL CDI(NROW2,NCOL2)
 155      CHARACTER CNAME*30,FNAME*24
 156         CHARACTER NN(NROW1)*30
 157
 158 C    INCREMENT PAGE COUNTER BY ONE
 159         IPGCTR=IPGCTR+1
 160
 161 C       START FIRST PAGE
 162 C       PRINT FORM FEED
 163          WRITE(IDEVNO,20,IOSTAT=IOVAL,ERR=900)
 164     020 FORMAT('1')
 165 C       PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 &8
 166           WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)
 167     040   FORMAT(1X,'')
 168      CALL HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)
 169           IF(NINC.EQ.0) THEN
 170             CALL HDG3(1,TN1,TN,IDEVNO)
 171           ELSEIF(NINC.EQ.-1) THEN
 172             CALL HDG3(2,TN1,TN,IDEVNO)
```

```
173           ELSE
174             CALL HDG2(NINC,TN1,TN,IDEVNO)
175           ENDIF
176           CALL HDG8(IDEVNO)
177 C       PRINT ANOTHER BLANK LINE
178           WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)
179
180 C       BEGIN LOOP FOR EACH CONTAMINANT 1 TO LIN
181         DO 100 I=1,LIN
182
183 C           PRINT 56 LINES OF DATA AND THEN START NEW PAGE
184           WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900) I,NN(I),
185    +CC(I,5),CC(I,8),CC(I,11),CC(I,14),CC(I,17),CC(I,20),CC(I,23),
186      +CC(I,26)
187     010  FORMAT(1X,I4,1X,A,8(1X,G11.4))
188
189 C     CHECK FOR 56 LINES-IF SO, INCREMENT PAGE NUMBER+START NEW PAGE
190           IF(INT(REAL(I)/56).EQ.REAL(I)/56) THEN
191             IPGCTR=IPGCTR+1
192 C           START SUBSEQUENT PAGES
193 C           PRINT FORM FEED
194             WRITE(IDEVNO,50,IOSTAT=IOVAL,ERR=900)
195     050     FORMAT('1')
196 C           PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 &8
197           WRITE(IDEVNO,30,IOSTAT=IOVAL,ERR=900)
198     030   FORMAT(1X,'')
199       CALL HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)
200           IF(NINC.EQ.0) THEN
201             CALL HDG3(1,TN1,TN,IDEVNO)
202           ELSEIF(NINC.EQ.-1) THEN
203             CALL HDG3(2,TN1,TN,IDEVNO)
204           ELSE
205             CALL HDG2(NINC,TN1,TN,IDEVNO)
206           ENDIF
207           CALL HDG8(IDEVNO)
208 C         PRINT ANOTHER BLANK LINE
209           WRITE(IDEVNO,30,IOSTAT=IOVAL,ERR=900)
210
211         ENDIF
212     100 CONTINUE
213
214         GO TO 999
215     900 WRITE(*,*)'IO ERROR IN PRREM1= ',IOVAL
216     999 RETURN
217         END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0

218
219 C       ************************************************************
220 C       *    SUBROUTINE PRREM2                                     *
221 C       * PROGRAM TO PRINT ANSWERS-RATE OF CONTAMINAMT REMOVAL (MG/HR) *
222 C       * SHEET 2                                                  *
223 C       ************************************************************
224
225 C       NOTES:(1)FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
226 C           (2)IDEVNO MUST BE 6 FOR FORM FEEDS TO BE PRINTED
227
228         SUBROUTINE PRREM2(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
```

```
229     +IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR) 230
231 C   SUBROUTINES REQUIRED:
232 C       HDG1,HDG2,HDG3,HDG9
233
234 C       TN,TN1=FINAL AND INITIAL INCREMENT TIME (HRS)
235 C       LIN=TOTAL NUMBER OF CONTAMINANTS
236 C       CC,NROW1,NCOL1=NAME & SIZE OF MAT CC
237 C       CDI,NROW2,NCOL2=NAME & SIZE OF MAT CDI
238 C       NN=NAME OF MAT NN
239 C       IDEVNO=DEVICE NUMBER FOR OUTPUT
240 C       NINC=TIME INCREMENT NUMBER
241 C         =0 THEN PRINT HDG3 WITH PCALC
242 C         =-1 THEN PRINT HDG3 WITH FINAL
243 C         ELSE PRINT HDG2 WITH INCREMENT NUMBER
244 C       IMONTH..IMINUTE=TIME AND DATE INFO
245 C       FNAME=FILE NAME OUTPUT DATA IS STORED ON
246 C       IPGCTR=COUNTER FOR SEQUENTIAL PAGE NUMBERS ON ALL OUTPUT
247
248         REAL CC(NROW1,NCOL1)
249         REAL CDI(NROW2,NCOL2)
250         CHARACTER CNAME*30,FNAME*24
251         CHARACTER NN(NROW1)*30
252
253 C       INCREMENT PAGE COUNTER BY ONE
254         IPGCTR=IPGCTR+1
255
256 C       START FIRST PAGE
257 C       DON'T PRINT FORM FEED UNLESS NO. CONT > 20
258         IF(LIN.GT.20) THEN
259 C         PRINT FORM FEED
260           WRITE(IDEVNO,20,IOSTAT=IOVAL,ERR=900)
261  020    FORMAT('1')
262         ENDIF
263 C       PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 &9
264           WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)
265  040    FORMAT(1X,'')
266           CALL HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)
267           IF(NINC.EQ.0) THEN
268            CALL HDG3(1,TN1,TN,IDEVNO)
269           ELSEIF(NINC.EQ.-1) THEN
270            CALL HDG3(2,TN1,TN,IDEVNO)
271           ELSE
272            CALL HDG2(NINC,TN1,TN,IDEVNO)
273           ENDIF
274           CALL HDG9(IDEVNO)
275 C       PRINT ANOTHER BLANK LINE
276           WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)
277
278 C       BEGIN LOOP FOR EACH CONTAMINANT 1 TO LIN
279         DO 100 I=1,LIN
280
281 C         PRINT 56 LINES OF DATA AND THEN START NEW PAGE
282           WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900) I,NN(I),
283    +CC(I,29),CC(I,32),CC(I,35),CC(I,38),CC(I,41),CC(I,44),CC(I,47)
284   010   FORMAT(1X,I4,1X,A,7(1X,G11.4))
285
286 C     CHECK FOR 56 LINES-IF SO, INCREMENT PAGE NUMBER+START NEW PAGE
287           IF(INT(REAL(I)/56).EQ.REAL(I)/56) THEN
288             IPGCTR=IPGCTR+1
289 C           START SUBSEQUENT PAGES
```


C-2.

```
290 C           PRINT FORM FEED
291              WRITE(IDEVNO,50,IOSTAT=IOVAL,ERR=900)
292  050         FORMAT('1')
293 C           PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 &8
294             WRITE(IDEVNO,30,IOSTAT=IOVAL,ERR=900)
295  030        FORMAT(1X,'')
296       CALL HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)
297             IF(NINC.EQ.0) THEN
298              CALL HDG3(1,TN1,TN,IDEVNO)
299             ELSEIF(NINC.EQ.-1) THEN
300              CALL HDG3(2,TN1,TN,IDEVNO)
301             ELSE
302              CALL HDG2(NINC,TN1,TN,IDEVNO)
303             ENDIF
304             CALL HDG9(IDEVNO)
305 C           PRINT ANOTHER BLANK LINE
306             WRITE(IDEVNO,30,IOSTAT=IOVAL,ERR=900)
307
308           ENDIF
309  100 CONTINUE
310
311         GO TO 999
312  900 WRITE(*,*)'IO ERROR IN PRREM2= ',IOVAL
313  999 RETURN
314         END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0

315
316 C       ************************************************************************
317 C       *    SUBROUTINE PRMAS1                                                 *
318 C       * PROGRAM TO PRINT ANSWERS-SUM OF CONT REMOVED BY DEVICE (MG)          *
319 C       * SHEET 1                                                             *
320 C       ************************************************************************
321
322 C    NOTES:(1)FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
323 C             (2)IDEVNO MUST BE 6 FOR FORM FEEDS TO BE PRINTED
324
325         SUBROUTINE PRMAS1(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
326    +IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR)
327
328 C   SUBROUTINES REQUIRED:
329 C       HDG1,HDG2,HDG3,HDG5
330
331 C    TN,TN1=FINAL AND INITIAL INCREMENT TIME (HRS)
332 C       LIN=TOTAL NUMBER OF CONTAMINANTS
333 C       CC,NROW1,NCOL1=NAME & SIZE OF MAT CC
334 C    CDI,NROW2,NCOL2=NAME & SIZE OF MAT CDI
335 C       NN=NAME OF MAT NN
336 C       IDEVNO=DEVICE NUMBER FOR OUTPUT
337 C       NINC=TIME INCREMENT NUMBER
338 C         =0 THEN PRINT HDG3 WITH PCALC
339 C         =-1 THEN PRINT HDG3 WITH FINAL
340 C         ELSE PRINT HDG2 WITH INCREMENT NUMBER
341 C       IMONTH..IMINUTE=TIME AND DATE INFO
342 C       FNAME=FILE NAME OUTPUT DATA IS STORED ON
343 C     IPGCTR=COUNTER FOR SEQUENTIAL PAGE NUMBERS ON ALL OUTPUT
344
345         REAL CC(NROW1,NCOL1)
```

```
346         REAL CDI(NROW2,NCOL2)
347         CHARACTER CNAME*30,FNAME*24
348         CHARACTER NN(NROW1)*30
349
350 C       INCREMENT PAGE COUNTER BY ONE
351         IPGCTR=IPGCTR+1
352
353 C       START FIRST PAGE
354 C       PRINT FORM FEED
355          WRITE(IDEVNO,20,IOSTAT=IOVAL,ERR=900)
356    020 FORMAT('1')
357 C       PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 &5
358           WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)
359    040  FORMAT(1X,'')
360           CALL HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)
361           IF(NINC.EQ.0) THEN
362             CALL HDG3(1,TN1,TN,IDEVNO)
363           ELSEIF(NINC.EQ.-1) THEN
364             CALL HDG3(2,TN1,TN,IDEVNO)
365           ELSE
366             CALL HDG2(NINC,TN1,TN,IDEVNO)
367           ENDIF
368           CALL HDG5(IDEVNO)
369 C       PRINT ANOTHER BLANK LINE
370           WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)
371
372 C       BEGIN LOOP FOR EACH CONTAMINANT 1 TO LIN
373         DO 100 I=1,LIN
374
375 C         PRINT 56 LINES OF DATA AND THEN START NEW PAGE
376           WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900) I,NN(I),
377      +CC(I,6),CC(I,9),CC(I,12),CC(I,15),CC(I,18),CC(I,21),CC(I,24),
378      +CC(I,27)
379    010  FORMAT(1X,I4,1X,A,8(1X,G11.4))
380
381 C         CHECK FOR 56 LINES-IF SO, INCREMENT PAGE NUMBER+START NEW PAGE
382           IF(INT(REAL(I)/56).EQ.REAL(I)/56) THEN
383             IPGCTR=IPGCTR+1
384 C           START SUBSEQUENT PAGES
385 C           PRINT FORM FEED
386              WRITE(IDEVNO,50,IOSTAT=IOVAL,ERR=900)
387    050       FORMAT('1')
388 C           PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 &8
389             WRITE(IDEVNO,30,IOSTAT=IOVAL,ERR=900)
390    030      FORMAT(1X,'')
391             CALL HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)
392             IF(NINC.EQ.0) THEN
393              CALL HDG3(1,TN1,TN,IDEVNO)
394             ELSEIF(NINC.EQ.-1) THEN
395              CALL HDG3(2,TN1,TN,IDEVNO)
396             ELSE
397              CALL HDG2(NINC,TN1,TN,IDEVNO)
398             ENDIF
399             CALL HDG5(IDEVNO)
400 C           PRINT ANOTHER BLANK LINE
401             WRITE(IDEVNO,30,IOSTAT=IOVAL,ERR=900)
402
403           ENDIF
404    100 CONTINUE
405
```

```
 406         GO TO 999
 407     900 WRITE(*,*)'IO ERROR IN PRMAS1= ',IOVAL
 408     999 RETURN
 409         END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0

 410
 411 C       ********************************************************************
 412 C       *    SUBROUTINE PRMAS2                                             *
 413 C       * PROGRAM TO PRINT ANSWERS-SUM OF CONT REMOVED BY DEVICE (MG)      *
 414 C       * SHEET 2                                                          *
 415 C       ********************************************************************
 416
 417 C    NOTES:(1)FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
 418 C            (2)IDEVNO MUST BE 6 FOR FORM FEEDS TO BE PRINTED
 419
 420         SUBROUTINE PRMAS2(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
 421    +IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR)
 422
 423 C   SUBROUTINES REQUIRED:
 424 C       HDG1,HDG2,HDG3,HDG6
 425
 426 C    TN,TN1=FINAL AND INITIAL INCREMENT TIME (HRS)
 427 C       LIN=TOTAL NUMBER OF CONTAMINANTS
 428 C       CC,NROW1,NCOL1=NAME & SIZE OF MAT CC
 429 C    CDI,NROW2,NCOL2=NAME & SIZE OF MAT CDI
 430 C       NN=NAME OF MAT NN
 431 C       IDEVNO=DEVICE NUMBER FOR OUTPUT
 432 C       NINC=TIME INCREMENT NUMBER
 433 C         =0 THEN PRINT HDG3 WITH PCALC
 434 C         =-1 THEN PRINT HDG3 WITH FINAL
 435 C         ELSE PRINT HDG2 WITH INCREMENT NUMBER
 436 C       IMONTH..IMINUTE=TIME AND DATE INFO
 437 C       FNAME=FILE NAME OUTPUT DATA IS STORED ON
 438 C    IPGCTR=COUNTER FOR SEQUENTIAL PAGE NUMBERS ON ALL OUTPUT
 439
 440         REAL CC(NROW1,NCOL1)
 441         REAL CDI(NROW2,NCOL2)
 442         CHARACTER CNAME*30,FNAME*24
 443         CHARACTER NN(NROW1)*30
 444
 445 C       INCREMENT PAGE COUNTER BY ONE
 446         IPGCTR=IPGCTR+1
 447
 448 C       START FIRST PAGE
 449 C    DON'T PRINT FORM FEED UNLESS NO. CONT > 20
 450         IF(LIN.GT.20) THEN
 451 C         PRINT FORM FEED
 452           WRITE(IDEVNO,20,IOSTAT=IOVAL,ERR=900)
 453   020   FORMAT('1')
 454         ENDIF
 455 C       PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 &9
 456           WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)
 457   040   FORMAT(1X,'')
 458      CALL HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)
 459           IF(NINC.EQ.0) THEN
 460            CALL HDG3(1,TN1,TN,IDEVNO)
 461           ELSEIF(NINC.EQ.-1) THEN
```

```
 462            CALL HDG3(2,TN1,TN,IDEVNO)
 463           ELSE
 464            CALL HDG2(NINC,TN1,TN,IDEVNO)
 465           ENDIF
 466           CALL HDG6(IDEVNO)
 467 C       PRINT ANOTHER BLANK LINE
 468           WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)
 469
 470 C       BEGIN LOOP FOR EACH CONTAMINANT 1 TO LIN
 471         DO 100 I=1,LIN
 472
 473 C         PRINT 56 LINES OF DATA AND THEN START NEW PAGE
 474           WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900) I,NN(I),
 475    +CC(I,30),CC(I,33),CC(I,36),CC(I,39),CC(I,42),CC(I,45),CC(I,48)
 476   010  FORMAT(1X,I4,1X,A,7(1X,G11.4))
 477
 478 C     CHECK FOR 56 LINES-IF SO, INCREMENT PAGE NUMBER+START NEW PAGE
 479           IF(INT(REAL(I)/56).EQ.REAL(I)/56) THEN
 480              IPGCTR=IPGCTR+1
 481 C            START SUBSEQUENT PAGES
 482 C            PRINT FORM FEED
 483               WRITE(IDEVNO,50,IOSTAT=IOVAL,ERR=900)
 484   050         FORMAT('1')
 485 C            PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 &8
 486              WRITE(IDEVNO,30,IOSTAT=IOVAL,ERR=900)
 487   030        FORMAT(1X,'')
 488         CALL HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)
 489              IF(NINC.EQ.0) THEN
 490               CALL HDG3(1,TN1,TN,IDEVNO)
 491              ELSEIF(NINC.EQ.-1) THEN
 492               CALL HDG3(2,TN1,TN,IDEVNO)
 493              ELSE
 494               CALL HDG2(NINC,TN1,TN,IDEVNO)
 495              ENDIF
 496              CALL HDG6(IDEVNO)
 497 C            PRINT ANOTHER BLANK LINE
 498              WRITE(IDEVNO,30,IOSTAT=IOVAL,ERR=900)
 499
 500           ENDIF
 501   100 CONTINUE
 502
 503         GO TO 999
 504   900 WRITE(*,*)'IO ERROR IN PRMAS2= ',IOVAL
 505   999 RETURN
 506         END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0

 507
 508 C       ********************************************************************
 509 C       *    SUBROUTINE PREFF                                              *
 510 C       * PROGRAM TO PRINT ANSWERS-END OF INCREMENT REMOVALL EFF (DEC)     *
 511 C       *                                                                  *
 512 C       ********************************************************************
 513
 514 C    NOTES:(1)FILE MUST BE OPEN BEFORE STARTING THIS SUBROUTINE
 515 C             (2)IDEVNO MUST BE 6 FOR FORM FEEDS TO BE PRINTED
 516
 517         SUBROUTINE PREFF(TN,TN1,LIN,CC,NROW1,NCOL1,CDI,NROW2,NCOL2,NN,
```

```
518    +IDEVNO,NINC,IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IOVAL,IPGCTR,
519    +PRTSW8,PRTSW9,IDEVN2)
520
521 C  SUBROUTINES REQUIRED:
522 C       HDG1,HDG2,HDG3,HDG7
523
524 C    TN,TN1=FINAL AND INITIAL INCREMENT TIME (HRS)
525 C       LIN=TOTAL NUMBER OF CONTAMINANTS
526 C       CC,NROW1,NCOL1=NAME & SIZE OF MAT CC
527 C    CDI,NROW2,NCOL2=NAME & SIZE OF MAT CDI
528 C       NN=NAME OF MAT NN
529 C       IDEVNO=DEVICE NUMBER FOR OUTPUT
530 C       NINC=TIME INCREMENT NUMBER
531 C         =0 THEN PRINT HDG3 WITH PCALC
532 C         =-1 THEN PRINT HDG3 WITH FINAL
533 C     ELSE PRINT HDG2 WITH INCREMENT NUMBER
534 C    IMONTH..IMINUTE=TIME AND DATE INFO
535 C       FNAME=FILE NAME OUTPUT DATA IS STORED ON
536 C       IPGCTR=COUNTER FOR SEQUENTIAL PAGE NUMBERS ON ALL OUTPUT
537
538         REAL CC(NROW1,NCOL1)
539         REAL CDI(NROW2,NCOL2)
540         CHARACTER CNAME*30,FNAME*24
541         CHARACTER NN(NROW1)*30
542         INTEGER PRTSW8,PRTSW9,IDEVN2,IDEVNO,I,J,K,H,NINC
543         IF ((PRTSW8.EQ.1).OR.((PRTSW8.EQ.0).AND.(NINC.EQ.-1))) THEN
544 C       INCREMENT PAGE COUNTER BY ONE
545         IPGCTR=IPGCTR+1
546
547 C       START FIRST PAGE
548 C       PRINT FORM FEED
549          WRITE(IDEVNO,20,IOSTAT=IOVAL,ERR=900)
550   020 FORMAT('1')
551 C       PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 &9
552           WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)
553   040   FORMAT(1X,'')
554           CALL HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)
555           IF(NINC.EQ.0) THEN
556             CALL HDG3(1,TN1,TN,IDEVNO)
557           ELSEIF(NINC.EQ.-1) THEN
558             CALL HDG3(2,TN1,TN,IDEVNO)
559           ELSE
560             CALL HDG2(NINC,TN1,TN,IDEVNO)
561           ENDIF
562           CALL HDG7(IDEVNO)
563 C       PRINT ANOTHER BLANK LINE
564           WRITE(IDEVNO,40,IOSTAT=IOVAL,ERR=900)
565
566 C       BEGIN LOOP FOR EACH CONTAMINANT 1 TO LIN
567         DO 100 I=1,LIN
568
569 C           PRINT 56 LINES OF DATA AND THEN START NEW PAGE
570           WRITE(IDEVNO,10,IOSTAT=IOVAL,ERR=900) I,NN(I),
571    +CC(I,7),CC(I,10),CC(I,13),CC(I,16),CC(I,19),CC(I,22),CC(I,25),
572    +CC(I,28),CC(I,31),CC(I,34),CC(I,37),CC(I,40),CC(I,43),CC(I,46)
573   010   FORMAT(1X,I4,1X,A,14(1X,F5.3))
574
575 C           CHECK FOR 56 LINES-IF SO, INCREMENT PAGE NUMBER+START NEW PAGE
576           IF(INT(REAL(I)/56).EQ.REAL(I)/56) THEN
577             IPGCTR=IPGCTR+1
```

```
578 C           START SUBSEQUENT PAGES
579 C           PRINT FORM FEED
580              WRITE(IDEVNO,50,IOSTAT=IOVAL,ERR=900)
581  050         FORMAT('1')
582 C           PRINT BLANK LINE FOLLOWED BY HEADINGS 1, 3 &8
583             WRITE(IDEVNO,30,IOSTAT=IOVAL,ERR=900)
584  030        FORMAT(1X,'')
585             CALL HDG1(IMONTH,IDAY,IYEAR,IHOUR,IMINUTE,FNAME,IPGCTR,IDEVNO)
586             IF(NINC.EQ.0) THEN
587              CALL HDG3(1,TN1,TN,IDEVNO)
588             ELSEIF(NINC.EQ.-1) THEN
589              CALL HDG3(2,TN1,TN,IDEVNO)
590             ELSE
591              CALL HDG2(NINC,TN1,TN,IDEVNO)
592             ENDIF
593             CALL HDG7(IDEVNO)
594 C           PRINT ANOTHER BLANK LINE
595             WRITE(IDEVNO,30,IOSTAT=IOVAL,ERR=900)
596
597           ENDIF
598  100 CONTINUE
599         ENDIF
600 C       *************** WRITE DATA TO A FILE FOR PLOTTING *************
601         IF (NINC.NE.-1) THEN
602         IF ((PRTSW9.EQ.2).OR.(PRTSW9.EQ.3)) THEN
603          DO 70 I=1,LIN,300
604           IS=I
605           IE=I+299
606           IF (IE.GT.LIN) IE=LIN
607           K=7
608           DO 60 H=2,15
609        WRITE (IDEVN2,55,IOSTAT=IOVAL,ERR=900) TN1,TN,H,
610       +        (CC(J,K),J=IS,IE)
611  055     FORMAT (T2,2(F8.2,1X),I2,1X,300(F5.3,:,1X))
612           K=K+3
613  060    CONTINUE
614  070   CONTINUE
615         ENDIF
616         ENDIF
617         GO TO 999
618  900 WRITE(*,*)'IO ERROR IN PREFF= ',IOVAL
619  999 RETURN
620         END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS  IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\PREDCT. Options: /C 80 /L /BY   05/21/92 12:58:48
C      ****************************************************************
C      *       SUBROUTINE PREDCT                                      *
C      * BASED ON REMOVAL EFF & SUM MASS REMOVED OF LAST INCREMENT,    *
C      * AND M.GEN OF THIS INCREMENT, PREDICT CAV PRED                 *
C      *   (CEQUILIB, CFINAL, M.REM ARE ALSO CALC, BUT NOT NEEDED)     *
C      *     WORKS FOR ONE CONT AT A TIME                              *
C      ****************************************************************

       SUBROUTINE PREDCT(I,TN,TN1,CAVPRD,DD,NROW,NCOL,
      +CC,NROW1,NCOL1,CDI,NROW2,NCOL2,CAVCLC,CFINAL,CEQLIB,LIN,LIN2,NN)

       INTEGER NROW,NCOL,NROW1,NCOL1,NROW2,NCOL2
       CHARACTER NN(NROW1)*30
       REAL DD(NROW,NCOL)
       REAL CC(NROW1,NCOL1)
       REAL CDI(NROW2,NCOL2)
C
C SUBROUTINES REQUIRED:
C PRAFIL-ZERO MAT DD COL 17-21
C LODEFF-LOAD REM EFF FOR LAST INCR FROM MAT CC INTO MAT DD COL 20
C MASBAL-CALC CAV PRED BASED ON REM EFF OF LAST INC & M.GEN OF THIS INC
C
C INPUTS:
C FROM MCALC
C    I=CONTAMINANT NO.
C    TN,TN1 =INCREMENT END & BEGINNING TIME (HRS)
C    DD,NROW,NCOL=NAME & DIM OF MAT DD
C    CC,NROW1,NCOL1=NAME & DIM OF MAT CC
C    CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
C    LIN=NO. OF CONTAMINANTS IN MAT CDI
C    LIN2=NO. DEVICES IN MAT DD
C    NN=NAME OF MAT NN
C FROM LODEFF
C    TAKES REM EFF FOR LAST INCR (IN MAT CC) AND PUTS IT IN
C      MAT DD COL 20 (FOR ALL DEVICES)
C FROM MASBAL
C    CAVCLC=CALC CABIN CONT CONC (MG/CU M)
C OUTPUTS:
C TO LODEFF
C    I=CONTAMINANT LINE NUMBER IN MAT CC
C    DD,NROW,NCOL=NAME & DIMENSIONS OF MAT DD
C    CC,NROW1,NCOL1=NAME & DIMENSIONS OF MAT CC
C    LIN2=NO. OF DEVICES IN MAT DD
C TO MASBAL
C    TN,TN1 =INCREMENT END & BEGINNING TIME (HRS)
C    CNVERR=CONVERGENCE ERROR
C    DD,NROW,NCOL=NAME & DIM OF MAT DD
C    CC,NROW1,NCOL1=NAME & DIM OF MAT CC
C    CDI,NROW2,NCOL2=NAME & DIM OF MAT CDI
C    LIN=NO. OF CONTAMINANTS IN MAT CDI
C    LIN2=NO. DEVICES IN MAT DD
C    CVOL=CABIN VOLUME (CU M)
C    TCABIN=CABIN TEMP (DEG K)
C    CINIT=INCR INIT CABIN CONT CONC (MG/CU M)=CC(I,1)
C TO MCALC
C    CAVCLC
C
C      ZERO MAT DD COL 17-21
```

```
   59          CALL PRAFIL(DD,NROW,NCOL,17,21)
   60
   61 C       LOAD REM EFF FROM LAST TIME INCR FROM MAT CC INTO MAT DD COL 20
   62          CALL LODEFF(I,DD,NROW,NCOL,CC,NROW1,NCOL1,LIN2)
   63
   64 C       FIND CAV PRED FOR THESE REMOVAL EFFICIENCIES
   65            CALL MASBAL(I,TN,TN1,DD,NROW,NCOL,CC,NROW1,NCOL1,
   66         +  CAVPRD,CDI,NROW2,NCOL2,CFINAL,CEQLIB,LIN,LIN2)
   67 C
   68 C       SET CAV IN PRED DD(I,22)=CAV IN CALC DD(I,17)
   69         DO 100 J=1,LIN2
   70          DD(J,22)=DD(J,17)
   71  100 CONTINUE
   72 C
   73         RETURN
   74 C       ************ END OF SUBROUTINE PREDCT  ***************************
   75         END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS  IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\RAFILL. Options: /C 80 /L /BY   05/21/92 12:58:55
   1 C      ************************************************************************
   2 C      *      SUBROUTINE RAFILL                                                *
   3 C      * SUBROUTINE TO FILL ADJUSTABLE SIZE REAL ARRAY WITH ZEROS              *
   4 C      ************************************************************************
   5        SUBROUTINE RAFILL(XX,NROW,NCOL)
   6        INTEGER NROW,NCOL
   7        REAL XX(NROW,NCOL)
   8
   9 C     XX=ARRAY NAME
  10 C     NCOL= COLUMNS IN MATRIX
  11 C     NROW= ROWS IN MATRIX
  12
  13         DO 110 I=1,NROW
  14         DO 100 J=1,NCOL
  15          XX(I,J)=0.0
  16 100 CONTINUE
  17 110 CONTINUE
  18         RETURN
  19 C       ************ END OF SUBROUTINE RAFILL ***************************
  20         END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS  IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\FORTRAN\TCC\RCHBD. Options: /C 80 /L /BIJY  03/15/94 10:41:07

  1 C   ******************************************************
  2 C   * FILE:RCHBD.FOR                                     *
  3 C   * SUBROUTINE FOR REM EFF-RADIAL FLOW CHARCOAL BED    *
  4 C   * DOESNT ALLOW FOR DESORPTION                        *
  5 C   * ASSUMES RELATIVELY THIN BED (OD CLOSE TO ID)       *
  6 C   ******************************************************
  7 C
  8     SUBROUTINE RCHBD (TN,TN1,CIIN,TCABIN,COEXIS,BEDQ,EMAX,CARTL,
  9    +BEDOD,BEDID,DENCH,TRTTYP,DCONT,VMOL,MW,VCONC,SOL,SMR,EFF,RH)
 10 C    OUTPUT:
 11 C     EFF=BED REMOVAL EFF(DEC)
 12 C    INPUTS:
 13 C     TN,TNI=INCREMENT INITIAL AND FINAL TIMES(HR)
 14 C     CIIN=BED INLET CONT CONC (MG/CU M)
 15 C     TCABIN=CABIN TEMP (DEG K)
 16 C     COEXIS=COEXISTANCE FACTOR
 17 C     BEDQ=BED FLOW RATE(CU M/HR)
 18 C     EMAX=MAXIMUM BED EFF (DEC)
 19 C     CARTL=CARTRIDGE LENGTH (M)
 20 C     BEDOD=BED OUTSIDE DIAMETER (M)
 21 C     BEDID=BED INSIDE DIAMETER (M)
 22 C     DENCH=DENSITY OF CHARCOAL IN BED (KG/CU M)
 23 C     TRTTYP=BED TREATMENT TYPE(1=CI CHAR,2=PHOS ACID, OTHER #=NONE)
 24 C     DCONT=CONT LIQUID DENSITY (GM/CC)
 25 C     VMOL=CONT MOLAR VOL(GM/CC)
 26 C     MW=CONT MOLECULAR WGT
 27 C     VCONC=CONT VAPOR CONCENTRATION AT TCABIN (MG/CU M)
 28 C     SOL=HENRY'S LAW CONSTANT FOR WATER SOLUBILITY
 29 C     (ATM/MOL FRACTION)
 30 C     SMR=SUM OF CONT MASS STORED IN BED(MG)-FROM LAST INCR
 31 C
 32     REAL LPREV, LAVN1, LUTIL, LIMM, LAVAV, LADS,MW
 33     INTEGER FACID,FCI
 34 C
 35 C    SET CIN=CIIN (THIS PREVENTS CIN FROM BEING PASSED BACK UP
 36 C        TO OTHER SUBROUTINES IF IT IS SET TO 1E-20)
 37     CIN=CIIN
 38
 39 C    BED TREATMENT LOGIC
 40 C     FACID=FLAG IF BED IS TREATED WITH PHOSPHORIC ACID (Y=1 N=0)
 41 C     FCI=FLAG FOR CI CHAR IN BED (REMOVES FORMALDAHYDE)
 42     IF (NINT(TRTTYP).EQ.2) THEN
 43      FACID=1
 44      FCI=0
 45     ELSEIF (NINT(TRTTYP).EQ.1) THEN
 46      FACID=0
 47      FCI=1
 48     ELSE
 49      FACID=0
 50      FCI=0
 51     ENDIF
 52 C
 53 C    TEST FOR NO BED FLOW(BEDQ=<0) OR TN-TN1<=0;BEDL,BEDDIA,DENCH=0
 54     IF((BEDQ.LE.0).OR.(TN-TN1.LE.0).OR.(CARTL.LE.0).OR.(BEDOD.LE.0)
 55    +.OR.(DENCH.LE.0)) THEN
 56      EFF=0
 57      GOTO 199
```

```
58      ENDIF
59 C     TEST FOR CI CHARCOAL AND FORMALDEHYDE(FCI=1 AND MW=30.03)
60      IF((MW.EQ.30.03).AND.(FCI.EQ.1)) THEN
61       CALL RCICH(EFF,EMAX,CARTL,BEDOD,BEDID,DENCH,SMR,BEDQ)
62       GOTO 199
63      ENDIF
64 C
65 C     TEST FOR AMMONIA AND PHOS ACID ON CHAR(FACID=1 AND MW=17.0 )
66      IF ((MW.EQ.17.0).AND.(FACID.EQ.1)) THEN
67       CALL RACCH(EFF,EMAX,CARTL,BEDOD,BEDID,DENCH,SMR)
68       GOTO 199
69      ENDIF
70 C
71 C     TEST FOR MOL VOL=0 (NO CHAR REMOVAL)
72      IF (VMOL.EQ.0) THEN
73       EFF=0
74       GOTO 199
75      ENDIF
76 C
77 C     CHARCOAL REMOVAL EFFICIENCY CALCULATION
78 C     BED LENGTH (M)-ASSUMES THIN BED
79      BEDL=(BEDOD-BEDID)/2
80      IF (BEDL.LT.0) BEDL=0
81 C     BED WGT (KG)
82      BEDWGT=DENCH*.785*(BEDOD**2-BEDID**2)*CARTL
83 C     SUPERFICIAL BED VEL(FT/MIN)
84      BEDVEL=BEDQ*.0348/((BEDOD+BEDID)*CARTL)
85 C     TEST FOR CIN TOO SMALL IN AVAL CALC
86      IF (CIN.LT.1E-20) CIN=1E-20
87      AVAL=(TCABIN/VMOL)*LOG10(VCONC/CIN)
88 C     ADS ZONE LENGTH FOR 90% REMOVAL (M)
89      LADS=AVAL*.000275*(BEDVEL/1.3)**.8
90 C     GET QI(CC LIQ CONT/GM CHAR)
91      CALL FQI(AVAL,QI,FACID,SOL,RH)
92 C     LENGTH OF BED PREVIOUSLY USED BY CONT AT THIS C INLET (M)
93      LPREV=SMR*1.0E-6*COEXIS*BEDL/(DCONT*BEDWGT*QI)
94 C     RATE OF BED USAGE (M BED/ MG CONT)
95      LIMM=1.0E-6*COEXIS*BEDL/(DCONT*BEDWGT*QI)
96 C     LENGTH OF BED AVAILABLE FOR ADS ZONE AT BEGINNING OF INCR (M)
97      LAVN1=BEDL-LPREV
98      IF (LAVN1.LT.0) LAVN1=0
99 C     FIX HERE IF DESORPTION IS DESIRED
100     IF (LAVN1/LADS.GT.20) THEN
101      EFFAV=EMAX
102     ELSE
103 C      INIT INCR EFF BASED ON C IN AND BED L AVAIL AT BEG OF INCR(DEC)
104      EFAVN1=EMAX*(1-EXP(-2.3025851*LAVN1/LADS))
105 C     LOOP FOR EFFICIENCY
106      EFFAV=EFAVN1
107      DO 399 J=1,10,1
108 C      LENGTH OF BED UTILIZ IN INCR (M)
109       LUTIL=CIN*BEDQ*EFFAV*(TN-TN1)*LIMM
110       IF (LUTIL.GT.LAVN1) THEN
111        GOTO 299
112       ELSE
113 C       AVERAGE BED LENGTH AVAIL (M)
114        LAVAV=LAVN1-LUTIL/2
115        IF ((LAVAV/LADS).GE.20) THEN
116         EFFAV=EMAX
117         GOTO 299
```

```
 118        ELSE
 119 C        AV EFF BASED ON AV BED L AVAIL (DEC)
 120         EFFAV=EMAX*(1-EXP(-2.3025851*LAVAV/LADS))
 121        ENDIF
 122       ENDIF
 123  399  CONTINUE
 124  299 ENDIF
 125 C    MAX EFF BASED ON C IN AND RATE OF BED USAGE (DEC)
 126     EFFMAX=LAVN1/(CIN*BEDQ* (TN-TN1)*LIMM)
 127     IF (EFFAV.GT.EFFMAX) EFFAV=EFFMAX
 128     IF (EFFAV.LT.0) EFFAV=0
 129     IF (EFFAV.GT.EMAX) EFFAV=EMAX
 130 C    EFF=ACTUAL EFF OUTPUT FROM SUBROUTINE
 131     EFF=EFFAV
 132 C    REMOVE THIS CHECK IF DESORPTION IS ADDED
 133  199 IF (EFF.LT.0) EFF=0
 134     IF (EFF.GT.EMAX) EFF=EMAX
 135     RETURN
 136     END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0

 137 C    ********************************************************
 138 C    * SUBROUTINE RACCH  - CALCULATES REMOVAL EFF           *
 139 C    * BED WITH NH3 AND 1.22 MILLIMOLE H3PO4 ON CHAR       *
 140 C    ********************************************************
 141     SUBROUTINE RACCH(EFF,EMAX,CARTL,BEDOD,BEDID,DENCH,SMR)
 142 C    OUTPUTS
 143 C     EFF=OUTPUT REMOVAL EFF (DEC)
 144 C    INPUTS
 145 C     EMAX=MAXIMUM BED REMOVAL EFF (DEC)
 146 C     CARTL=CARTRIDGE LENGTH (M)
 147 C     BEDOD=BED OUTSIDE DIAMETER (M)
 148 C     BEDID=BED INSIDE DIAMETER (M)
 149 C     DENCH=CHARCOAL DENSITY(KG/CU M)
 150 C     SMR=SUM OF MASS OF CONT REMOVED AT BEG OF INCR (MG)
 151 C
 152 C    FOR AMMONIA CAPACITY AT SMAC
 153 C    CHAR USED (KG)
 154      CHRUSD=1.6E-4*SMR
 155 C    CHAR BED WGT(KG)
 156     BEDWGT=CARTL*(BEDOD**2-BEDID**2)*.785*DENCH
 157     IF (CHRUSD.LT.0.8*BEDWGT) THEN
 158      EFF=EMAX
 159     ELSE
 160      EFF=EMAX*SIN((BEDWGT-CHRUSD)*1.57/(BEDWGT*0.2))
 161     ENDIF
 162 C    PREVENTS NEGATIVE EFF FOR REACTION
 163 C    IF (EFF.LT.0) EFF=0
 164     IF(EFF.GT.EMAX) EFF=EMAX
 165     RETURN
 166     END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0
```

```
C    ******************************************************
C    * SUBROUTINE RCICH  - CALCULATES REMOVAL EFF          *
C    *  FOR FORMALDELYDE AND CI CHAR BED                   *
C    ******************************************************
     SUBROUTINE RCICH(EFF,EMAX,CARTL,BEDOD,BEDID,DENCH,SMR,BEDQ)
C    OUTPUTS
C     EFF=OUTPUT REMOVAL EFF (DEC)
C    INPUTS
C     EMAX=MAXIMUM BED REMOVAL EFF (DEC)
C     CARTL=CARTRIDGE LENGTH (M)
C     BEDOD=BED OUTSIDE DIAMETER (M)
C     BEDID=BED INSIDE DIAMETER (M)
C     DENCH=CHARCOAL DENSITY(KG/CU M)
C     SMR=SUM OF MASS OF CONT REMOVED AT BEG OF INCR (MG)
C     BEDQ=BED FLOW RATE (CU M/HR)
C
     BEDWGT=CARTL*(BEDOD**2-BEDID**2)*.785*DENCH
C     PERCENT OF BED WEIGHT CONSUMED (DEC)
     PBWGT=SMR/(BEDWGT*1E6)
     IF(PBWGT.LT..0012) THEN
      EFF=1-PBWGT*83.3
     ELSE
      EFF=.9*COS(PBWGT*1.57/.05)
     ENDIF
C     BED RESIDENCE TIME (SEC)
     BREST=(BEDOD-BEDID)*CARTL*(BEDOD+BEDID)*3600/(BEDQ*1.273)
     IF(BREST.LT.0.25)THEN
      EFF=EFF*BREST/.25
     ENDIF
C     PREVENTS NEGATIVE EFF FOR REACTION
C     IF (EFF.LT.0) EFF=0
     IF(EFF.GT.EMAX) EFF=EMAX
     RETURN
     END
```

```
NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS  IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\REGEN.F Options: /C 80 /L /BY   05/21/92 13:00:29
   1 C      ***************************************************************
   2 C      *       REGENERATION SUBROUTINE-REGEN                         *
   3 C      * FOR ALL BEDS (3-15) DETERMINES IF BED IS TO BE REGENERATED   *
   4 C      * AT BEGINNING OF TIME INCREMENT, AND IF REGENERATION IS TO    *
   5 C      * DURING THE ENTIRE TIME INCREMENT- IF THE BED IS TO BE        *
   6 C      * REGENERATED THE MASSES STORED ARE SET TO ZERO, AND IF        *
   7 C      * REGENERATION IS TO OCCUR THROUGHOUT THE TIME INCREMENT THE   *
   8 C      * BED FLOW RATE IS SET TO ZERO; OTHERWISE IT IS SET TO THE     *
   9 C      * ORIGINAL VALUE                                              *
  10 C      ***************************************************************
  11        SUBROUTINE REGEN(TN,TN1,DD,NROW,NCOL,CC,NROW1,NCOL1,
  12       +CDI,NROW2,NCOL2,LIN,LIN2,IMSGDN)
  13        INTEGER NROW,NCOL,NROW1,NCOL1,NROW2,NCOL2
  14        REAL DD(NROW,NCOL)
  15        REAL CC(NROW1,NCOL1)
  16        REAL CDI(NROW2,NCOL2)
  17
  18 C NOTE: BEFORE RUNNING THIS SUBROUTINE THE ORIGINAL FLOW RATES FROM 19 C
  20 C         TIME INCREMENT THEY MUST BE RESTORED TO COL 2
  21 C DIRECT INPUTS:
  22 C    TN=INCREMENT END TIME(HRS); TN1=INCR BEGINNING TIME HRS
  23 C    DD,NROW,NCOL=NAME AND SIZE OF MAT DD
  24 C    CC,NROW1,NCOL1=NAME AND SIZE OF MAT CC
  25 C    CDI,NROW2,NCOL2=NAME AND SIZE OF MAT CDI
  26 C    LIN=NO. OF CONT IN MAT CDI
  27 C    LIN2=NO. OF DEVICES IN MAT DD
  28
  29 C OTHER INPUTS FROM MAT DD
  30 C    TIR=INITIAL (FIRST) REGENERATION TIME (HRS)
  31 C    TRCI=REGEN/CHANGEOUT INTERVAL (HRS)
  32 C    TRD=REGENERATION DURATION (HRS)
  33 C    DEVICE NO., TYPE, FLOW RATE, ETC
  34
  35 C OUTPUTS:
  36 C    A)IF REGENERATION OCCURS AT THE BEGINNING OF ANY TIME INCREMENT
  37 C      1)FOR ANY DEVICE WHICH IS A CHARCOAL BED
  38 C        FOR ALL CONT 1 TO LIN IT PUTS SUM MASS REM=0 IN MAT CC
  39 C        COL 12,15,18....48 AS APPROPRIATE FOR THAT DEVICE
  40 C   2)FOR ANY DEVICE WHICH IS A LIOH BED
  41 C        IT DOES 1) ABOVE, AND IN ADDITION PUTS SUM MASS REM=0 IN
  42 C        MAT DD COL 16 FOR THAT DEVICE
  43 C  B)IF REGENERATION IS OCCURRING THROUGHOUT THE WHOLE INTERVAL
  44 C        IT SETS Q OF DEVICE=0; IF REGENERATION IS NOT OCCURRING, IT
  45 C        SETS Q=THE ORIGINAL VALUE
  46 C
  47 C   SUBROUTINES REQUIRED:
  48 C        REGCHG
  49 C
  50 C  START LOOP FOR ALL DEVICES 3 TO 15
  51    DO 100 J=3,LIN2
  52 C        IF DEVICE DOES NOT EQUAL CHARCOAL OR LIOH THEN GO TO END OF LOOP
  53          IF(DD(J,3).NE.3.AND.DD(J,3).NE.4.AND.DD(J,3).NE.5) GOTO 100
  54
  55 C   ASSIGN PROPER VARIABLES FOR DEVICE
  56 C        DEVICE = CHARCOAL
  57          IF(DD(J,3).EQ.3.OR.DD(J,3).EQ.4) THEN
  58 C          REGENERATION/CHANGEOUT INTERVAL (HRS)
  59            TRCI=DD(J,15)
```

```
60 C           REGENERATION DURATION (HRS)
61             TRD=DD(J,16)
62 C           INITIAL(FIRST) REGENERATION
63             TIR=DD(J,14)
64           ENDIF
65
66 C         DEVICE = LIOH
67           IF(DD(J,3).EQ.5) THEN
68 C           REGENERATION/CHANGEOUT INTERVAL (HRS)
69             TRCI=DD(J,14)
70 C           REGENERATION DURATION (HRS)
71             TRD=0
72 C           INITIAL(FIRST) REGENERATION
73             TIR=DD(J,13)
74           ENDIF
75
76 C    CHECK AND FIX INPUT AS REQ + PRINT WARNINGS
77 C         TIME INCREMENT (HRS)
78             TINC=DD(1,11)
79 C         INITIAL TIME NOT EQUAL TO MULTIPLE OF TIME INCREMENT
80             IF(AINT(TIR/TINC).NE.(TIR/TINC)) THEN
81               TIR=AINT(TIR/TINC)*TINC
82               OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
83                WRITE(IMSGDN,*)'INCREMENT BEGINNING TIME ',TN1,
84    +'DEV NO.',(J)
85                WRITE(IMSGDN,*)'INITIAL TIME NOT = MULTIPLE OF TIME
86    + INCREMENT'
87                WRITE(IMSGDN,*)'TRUNCATED TO ',TIR
88               CLOSE(IMSGDN)
89              ENDIF
90
91 C         REGEN/CHGOUT INTERVAL < TIME INCR OR NOT= MULTIPLE OF TIME INCR
92             IF(AINT(TRCI/TINC).NE.(TRCI/TINC)) THEN
93               TRCI=AINT(TRCI/TINC)*TINC
94               OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
95                WRITE(IMSGDN,*)'INCREMENT BEGINNING TIME ',TN1,
96    +'DEV NO.',(J)
97                WRITE(IMSGDN,*)'REGEN/CHG TIME NOT = MULTIPLE OF TIME
98    + INCREMENT'
99                WRITE(IMSGDN,*)'TRUNCATED TO ',TRCI
100              CLOSE(IMSGDN)
101             ENDIF
102
103 C          IF REGEN/CHGOUT INTERVAL <= O THEN GOTO END OF LOOP FOR CONT
104            IF (TRCI.LE.0) THEN
105              GOTO 100
106       ENDIF
107
108 C      REGEN DURATION < OR NOT = MULTIPLE OF TIME INCREMENT
109              IF(AINT(TRD/TINC).NE.(TRD/TINC)) THEN
110                TRD=AINT(TRD/TINC)*TINC
111                OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
112                  WRITE(IMSGDN,*)'INCREMENT BEGINNING TIME ',TN1,
113    +'DEV NO.',(J)
114         WRITE(IMSGDN,*)'REG DURATION NOT = MULTIPLE OF TIME
115    + INCREMENT'
116                  WRITE(IMSGDN,*)'TRUNCATED TO ',TRD
117                CLOSE(IMSGDN)
118              ENDIF
119
```

```
 120 C     REGENERATION DURATION > REGEN/CHG INTERVAL
 121               IF(TRD.GT.TRCI) THEN
 122                 TRD=TRCI
 123                 OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
 124                   WRITE(IMSGDN,*)'INCREMENT BEGINNING TIME ',TN1,
 125    +'DEV NO.',(J)
 126          WRITE(IMSGDN,*)'REGEN DURATION > REGEN/CHG INTERVAL'
 127                   WRITE(IMSGDN,*)'TRUNCATED TO ',TRD
 128                 CLOSE(IMSGDN)
 129               ENDIF
 130
 131 C    CHECK TO SEE IF REGENERATION OCCURS AT BEGINNING OF TIME INCR,
 132 C     AND IF REGEN OCCURS THROUGHOUT WHOLE TIME INCREMENT
 133
 134     CALL REGCHG(TN1,TRCI,TRD,TIR,TINC,IRBFLG,IRTFLG)
 135 C     REGENERATION OCCURS AT BEGINNING OF INCREMENT
 136               IF (IRBFLG.EQ.1) THEN
 137 C       PUT SUM MASS REM =0 IN MAT CC FOR THIS DEVICE
 138 C               START LOOP FOR ALL CONT FOR THIS DEVICE
 139                 K=J*3+3
 140                 DO 101 I=1,LIN
 141                   CC(I,K)=0
 142  101            CONTINUE
 143
 144 C       IF DEVICE = LIOH BED PUT SUM MASS=0 IN DD(J,16)
 145                 IF (DD(J,3).EQ.5) THEN
 146                   DD(J,16)=0
 147                 ENDIF
 148               ENDIF
 149
 150 C     REGENERATION OCCURS THROUGHOUT ENTIRE INCREMENT
 151               IF (IRTFLG.EQ.1) THEN
 152 C               SET DEVICE Q=0
 153                 DD(J,2)=0
 154               ELSE
 155 C               SET DEVICE Q= ORIGINAL VALUE
 156                 DD(J,2)=DD(J,7)
 157               ENDIF
 158
 159 C  END OF J LOOP FOR EACH DEVICE
 160  100 CONTINUE
 161
 162     RETURN
 163     END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0

 164
 165
 166 C    ****************************************************************
 167 C    *     AUXILIARY REGENERATION SUBROUTINE-REGCHG                 *
 168 C    * DETERMINES IF REGEN/CHANGEOUT IS TO OCCUR AT BEGINNING OF     *
 169 C    * TIME INCREMENT-ALSO DETERMINES IF REGENERATION IS OCCURRING   *
 170 C    * THROUGHOUT THE TIME INCREMENT                                *
 171 C    ****************************************************************
 172          SUBROUTINE REGCHG(TN1,TRCI,TRD,TIR,TINC,IRBFLG,IRTFLG)
 173 C
 174 C INPUTS:
 175 C     TN1=INCREMENT INITIAL TIME (HRS)
```

```
176 C    TRCI=CHANGEOUT/REGENERATION INTERVAL (HRS)
177 C    TRD=REGENERATION DURATION (HRS)
178 C    TIR=INITIAL (FIRST) REGENERATION TIME (HRS)
179 C    TINC=TIME INCREMENT (HRS)
180 C
181 C OUTPUTS:
182 C    REGENERATION OCCURS AT BEGINNING OF TIME INCREMENT (Y OR N)
183 C       (IRBFLG=1 FOR Y & 0 FOR N)
184 C  REGENERATION IS OCCURRING THROUGHOUT THE WHOLE INCREMENT (Y OR N)
185 C       (IRTFLG=1 FOR Y & 0 FOR N)
186 C
187
188 C       REGENERATION OCCURS AT BEGINNING OF TIME INCREMENT
189
190         IF (TN1.EQ.0) GOTO 10
191         IF(TN1.LT.TIR) GOTO 10
192         IF(TRCI.LE.0) GOTO 10
193     IF(AINT((TN1-TIR)/TRCI).EQ.((TN1-TIR)/TRCI)) THEN
194 C         REGENERATION OCCURS
195           IRBFLG=1
196           GO TO 20
197         ENDIF
198 C       NO REGENERATION OCCURS
199   010 IRBFLG=0
200   020 CONTINUE
201
202 C    REGENERATION OCCURRING THROUGHOUT ENTIRE TIME INCREMENT
203
204         IF(TRCI.LE.0) GOTO 30
205         IF((TRD.LE.0).OR.(TN1.LT.TIR)) GOTO 30
206         IF(TN1.GE.AINT((TN1-TIR)/TRCI)*TRCI+TIR+TRD) THEN
207           GOTO 30
208         ELSE
209 C         REGENERATION OCCURS
210           IRTFLG=1
211           GOTO 40
212         ENDIF
213
214 C       REGEN DOESN'T OCCUR
215   030 IRTFLG=0
216
217   040 CONTINUE
218         RETURN
219         END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS  IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\RINCDD. Options: /C 80 /L /BY   05/21/92 13:00:48
     1 C     ************************************************************************
     2 C     *       SUBROUTINE RINCDD                                              *
     3 C     *    SUBROUTINE TO OPERATE ON INCREMENT DEPENDENT DATA                 *
     4 C     * READS DATA FROM MAT TT AND PUT IT IN THE PROPER PLACES IN             *
     5 C     * MAT CDI OR MAT DD - USED AT THE BEGINNING OF EACH TIME INCR           *
     6 C     ************************************************************************
     7
     8       SUBROUTINE RINCDD(I,TN,TN1,DD,NROW,NCOL,LIN2,
     9    +CC,NROW1,NCOL1,CDI,NROW2,NCOL2,LIN,TT,NTTROW,NTTCOL,LIN1)
    10       INTEGER NROW,NCOL,NROW1,NCOL1,NROW2,NCOL2,NTTROW,NTTCOL
    11       REAL DD(NROW,NCOL)
    12       REAL CC(NROW1,NCOL1)
    13       REAL CDI(NROW2,NCOL2)
    14       REAL TT(NTTROW,NTTCOL)
    15
    16 C SUBROUTINES REQUIRED:
    17 C     NONE
    18 C     DD,NROW,NCOL,LIN2=NAME,DIM & NO DEV IN MAT DD
    19 C     CC,NROW1,NCOL1=NAME & DIM OF MAT CC
    20 C     CDI,NROW2,NCOL,LIN=NAME,DIM & NO CONT IN MAT CDI
    21 C     TT,NTTROW,NTTCOL,LIN1=NAME,DIM & NO ITEMS IN MAT TT
    22
    23       IF (LIN1.EQ.0) GOTO 999
    24 C     BEGIN LOOP FOR ALL LINES IN MAT TT
    25       DO 100 K=1,LIN1
    26
    27 C     CHECK FOR TIME >= TN1 AND < TN
    28       IF((TT(K,1).LT.TN1).OR.(TT(K,1).GE.TN)) THEN
    29        GO TO 100
    30       ENDIF
    31
    32 C     IDENTIFY VARIABLES
    33       ICONTN=NINT(TT(K,2))
    34       GENRT=TT(K,3)
    35       IDEVNO=NINT(TT(K,4))
    36       DEVQ=TT(K,5)
    37       ICOLNO=TT(K,6)
    38       VAL=TT(K,7)
    39
    40 C     ICONTN=INTEGER CONTAMINANT NO. - TT(K,2)
    41 C     GENRT=CONT GENERATION RATE (MG/HR) - TT(K,3)
    42 C     IDEVNO=INTEGER DEVICE NUMBER - TT(K,4)
    43 C     DEVQ=DEVICE FLOW RATE (CU M/HR) - TT(K,5)
    44 C     ICOLNO=INTEGER COLUMN NUMBER IN MAT DD - TT(K,6)
    45 C     VAL=NEW VALUE IN MAT DD - TT(K,7)
    46
    47 C     CASE NO. 1 - CHANGE CONTAMINANT GENERATION RATE
    48
    49       IF((ICONTN.GT.0).AND.(ICONTN.LE.LIN)) THEN
    50        IF(IDEVNO.EQ.1) THEN
    51         CDI(ICONTN,1)=GENRT
    52        ELSEIF ((IDEVNO.GE.3).AND.(IDEVNO.LE.LIN2)) THEN
    53         CDI(ICONTN,(7+IDEVNO))=GENRT
    54        ENDIF
    55        GOTO 100
    56       ENDIF
    57
    58 C     CASE 2 - CHANGE DEVICE FLOW OR OTHER DD DATA
```

```
 59 C     THIS CASE WORKS ONLY IF ANY CONT NO. <=0
 60 C     MUST USE -1 FOR ANY Q OR NEW VALUE NOT TO BE CHANGED
 61 C     MAT DD COL NO. <0 ALSO STOPS NEW VALUE FROM BEING CHANGED
 62
 63       IF(ICONTN.LE.0) THEN
 64        IF((IDEVNO.GE.1).AND.(IDEVNO.LE.LIN2)) THEN
 65         IF(DEVQ.GE.0) THEN
 66 C         CHANGE DEVICE FLOW IN MAT DD
 67           DD(IDEVNO,2)=DEVQ
 68         ENDIF
 69         IF((ICOLNO.GE.1).AND.(ICOLNO.LE.16)) THEN
 70          IF (VAL.GE.0) THEN
 71 C          CHANGE VALUE IN MAT DD
 72           DD(IDEVNO,ICOLNO)=VAL
 73          ENDIF
 74         ENDIF
 75        ENDIF
 76       ENDIF
 77   100 CONTINUE
 78   999 RETURN
 79       END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS  IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\RRIN.FO Options: /C 80 /L /BY   05/21/92 13:00:57
   1 C      ************************************************************************
   2 C      *     SUBROUTINE  RRIN                                                 *
   3 C      *  SUBROUTINE TO READ REAL DATA INTO MAT XX(ROW,COL)                   *
   4 C      *  RETURNS NUMBER OF LINES OF DATA READ FROM FILE                      *
   5 C      *  READS FROM COL 1 TO COL LSTCOL                                      *
   6 C      ************************************************************************
   7 C  NOTE: INPUT NUMBERS MUST BE SEPARATED BY BLANKS
   8        SUBROUTINE RRIN(XX,NROW,NCOL,LSTCOL,LIN)
   9        INTEGER NROW,NCOL,IOVAL,LSTCOL,LIN
  10        CHARACTER FNAME*24
  11        REAL XX(NROW,NCOL)
  12        IF(LSTCOL.GT.NCOL) LSTCOL=NCOL
  13    010 READ(*,'(A)') FNAME
  14        OPEN(1,FILE=FNAME,STATUS='OLD',IOSTAT=IOVAL)
  15        IF(IOVAL.NE.0) GOTO 900
  16        LIN=0
  17        DO 100 I=1,NROW
  18        READ(1,*,IOSTAT=IOVAL,END=500,ERR=900 ) (XX(I,J),J=1,LSTCOL)
  19        LIN=LIN+1
  20    100 CONTINUE
  21    500 WRITE(*,'(A)') ' DONE WITH FILE INPUT'
  22        WRITE (*,*) ' '
  23        CLOSE (1)
  24        GOTO 990
  25    900 WRITE(*,*)'IOERROR= ',IOVAL
  26        CLOSE (1)
  27        WRITE(*,*) 'WHAT IS THE INPUT FILE NAME? '
  28        GOTO 10
  29    990 RETURN
  30 C      ************ END OF SUBROUTINE RRIN *********************************
  31        END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS  IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\RROUT.F Options: /C 80 /L /BY   05/21/92 13:01:03
   1 C      ****************************************************************
   2 C      *    SUBROUTINE RROUT                                          *
   3 C      *  SUBROUTINE TO WRITE DATA TO CONSOLE, OR PRINTER             *
   4 C      *  WRITES REAL DATA FROM MAT XX(ROW,COL)                       *
   5 C      *  WRITES FROM FSTCOL TO LSTCOL                                *
   6 C      ****************************************************************
   7         SUBROUTINE RROUT(XX,NROW,NCOL,FSTCOL,LSTCOL,LIN,IMSGDN,FNAME,
   8        + IDEVNO,IOVAL)
   9         INTEGER NROW,NCOL,IOVAL,FSTCOL,LSTCOL,LIN,IDEVNO
  10         CHARACTER FNAME*24,DES*1
  11         REAL XX(NROW,NCOL)
  12         IF (FSTCOL.GT.NCOL) FSTCOL=NCOL
  13         IF (LSTCOL.GT.NCOL) LSTCOL=NCOL
  14         IF (FSTCOL.GT.LSTCOL) FSTCOL=LSTCOL
  15 C 010 OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
  16 C       WRITE(IMSGDN,'(A)') ' WRITE TO LPT1 OR CON OR END '
  17 C       CLOSE(IMSGDN)
  18 C       READ(*,'(A)') FNAME
  19 C       QUIT IF FNAME=END
  20 C       IF(FNAME.EQ.'END') GOTO 990
  21 C       IF((FNAME.NE.'LPT1').AND.(FNAME.NE.'CON')) GOTO 10
  22 C       OPEN(1,FILE=FNAME, IOSTAT=IOVAL)
  23         IF(IOVAL.NE.0) GOTO 900
  24         DO 110 I=1,LIN
  25 C        WRITE(1,70,IOSTAT=IOVAL,ERR=900) (XX(I,J),J=FSTCOL,LSTCOL)
  26          WRITE(IDEVNO,70,IOSTAT=IOVAL,ERR=900) (XX(I,J),J=FSTCOL,LSTCOL)
  27  070   FORMAT(1X,7G11.4)
  28 C        WRITE(1,*,IOSTAT=IOVAL,ERR=900)
  29          WRITE(IDEVNO,*,IOSTAT=IOVAL,ERR=900)
  30  110 CONTINUE
  31 C       CLOSE (1)
  32         GOTO 990
  33  900 OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
  34         WRITE(IMSGDN,*)'IOERROR= ',IOVAL
  35         CLOSE(IMSGDN)
  36 C       CLOSE (1)
  37         CLOSE (IDEVNO)
  38 C       GOTO 10
  39  990 RETURN
  40 C       *************** END OF SUBROUTINE RROUT ************************
  41         END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS  IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\RROUT2. Options: /C 80 /L /BY   05/21/92 13:01:08
   1 C      ************************************************************************
   2 C      *    SUBROUTINE RROUT2                                                 *
   3 C      *  SUBROUTINE TO WRITE DATA TO CONSOLE, OR PRINTER                     *
   4 C      *  WRITES REAL DATA FROM MAT XX(ROW,COL)                               *
   5 C      *  WRITES FROM FSTCOL TO LSTCOL                                        *
   6 C      ************************************************************************
   7        SUBROUTINE RROUT2 (XX,NROW,NCOL,FSTCOL,LSTCOL,LIN,IMSGDN)
   8        INTEGER NROW,NCOL,IOVAL,FSTCOL,LSTCOL,LIN,IDEVNO
   9        CHARACTER FNAME*24,DES*1
  10        REAL XX(NROW,NCOL)
  11        IF (FSTCOL.GT.NCOL) FSTCOL=NCOL
  12        IF (LSTCOL.GT.NCOL) LSTCOL=NCOL
  13        IF (FSTCOL.GT.LSTCOL) FSTCOL=LSTCOL
  14  010 OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
  15        WRITE(IMSGDN,'(A)') ' WRITE TO LPT1 OR CON OR END '
  16        CLOSE(IMSGDN)
  17        READ(*,'(A)') FNAME
  18        QUIT IF FNAME=END
  19        IF(FNAME.EQ.'END') GOTO 990
  20        IF((FNAME.NE.'LPT1').AND.(FNAME.NE.'CON')) GOTO 10
  21        OPEN(1,FILE=FNAME, IOSTAT=IOVAL)
  22        IF(IOVAL.NE.0) GOTO 900
  23        DO 110 I=1,LIN
  24         WRITE(1,70,IOSTAT=IOVAL,ERR=900) (XX(I,J),J=FSTCOL,LSTCOL)
  25  070   FORMAT(1X,7G11.4)
  26         WRITE(1,*,IOSTAT=IOVAL,ERR=900)
  27  110 CONTINUE
  28        CLOSE (1)
  29        GOTO 990
  30  900 OPEN(IMSGDN,FILE='CON',IOSTAT=IOVAL)
  31        WRITE(IMSGDN,*)'IOERROR= ',IOVAL
  32        CLOSE(IMSGDN)
  33        CLOSE (1)
  34        GOTO 10
  35  990 RETURN
  36 C      *************** END OF SUBROUTINE RROUT ************************
  37        END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
NUMBER OF ERRORS  IN COMPILATION : 0
```

```
RM/FORTRAN Compiler (V2.42)
Source File: C:\RMFORT\TCC\SLIOH.F Options: /C 80 /L /BY   05/21/92 13:01:15
   1 C      ***************************************************************
   2 C      *           SUBROUTINE-SLIOH                                  *
   3 C      * SUM LIOH USED IN TIME INCREMENT FOR EACH BED ONE AT A TIME   *
   4 C      * AND FOR ALL CONTAMINANTS FOR EACH BED                       *
   5 C      ***************************************************************
   6
   7        SUBROUTINE SLIOH(TN,TN1,DD,NROW,NCOL,CC,NROW1,NCOL1,
   8       +CDI,NROW2,NCOL2,LIN,LIN2)
   9        INTEGER NROW,NCOL,NROW1,NCOL1,NROW2,NCOL2
  10        REAL DD(NROW,NCOL)
  11        REAL CC(NROW1,NCOL1)
  12        REAL CDI(NROW2,NCOL2)
  13 C
  14 C SUBROUTINES REQUIRED: NONE
  15 C DIRECT INPUTS:
  16 C    TN=INCREMENT END TIME(HRS); TN1=INCR BEGINNING TIME HRS
  17 C    DD,NROW,NCOL=NAME AND SIZE OF MAT DD
  18 C    CC,NROW1,NCOL1=NAME AND SIZE OF MAT CC
  19 C    CDI,NROW2,NCOL2=NAME AND SIZE OF MAT CDI
  20 C    LIN=NO. OF CONT IN MAT CDI
  21 C    LIN2=NO. OF DEVICES IN MAT DD
  22
  23 C OTHER INPUTS FROM MAT DD
  24 C    DD(J,3)=DEVICE NUMBER
  25 C    DD(J,16)=AMT OF LIOH PREVIOUSLY USED BY DEVICE
  26 C    CDI(I,17)=LB LIOH UTIL/LB CONT ADSORBED IN BED (FOR ONE CONT)
  27
  28 C OUTPUTS (STORED IN MAT DD):
  29 C    DD(J,16)=AMOUNT OF LIOH UTILIZED BY DEVICE THROUGH THE END OF
  30 C    THIS TIME INCREMENT
  31 C    DD(J,15)=RATE OF LIOH USAGE FOR DEVICE
  32 C
  33         K=11
  34
  35 C       START LOOP FOR ALL DEVICES 3 TO 15
  36         DO 100 J=3,LIN2
  37 C        CHECK FOR DEVICE = LIOH BED
  38          IF (DD(J,3).EQ.5) THEN
  39 C         RATE OF LIOH UTILIZATION (KG/HR)
  40           RWUTLI=0
  41 C         BEGIN LOOP FOR ALL CONTAMINANTS
  42           DO 110 I=1,LIN
  43            RWUTLI=RWUTLI+CC(I,K)*CDI(I,7)*1E-6
  44  110     CONTINUE
  45 C          STORE RATE OF LIOH UTILIZATION IN MAT DD FOR THIS DEVICE
  46           DD(J,15)=RWUTLI
  47 C       UPDATE AMOUNT OF LIOH UTIL THROUGH THE END OF TIME INCR(KG)
  48          DD(J,16)=DD(J,16)+RWUTLI*(TN-TN1)
  49          ENDIF
  50          K=K+3
  51 C     END J LOOP
  52   100 CONTINUE
  53        RETURN
  54        END

NUMBER OF WARNINGS IN PROGRAM UNIT: 0
NUMBER OF ERRORS  IN PROGRAM UNIT: 0
NUMBER OF WARNINGS IN COMPILATION : 0
```

```
NUMBER OF ERRORS  IN COMPILATION : 0
```

APPENDIX B

TOXIC HAZARD INDEX DESCRIPTION

The toxic hazard index, or T-value, is the method used by toxicologists to assess the acceptability of an atmosphere containing a mixture of contaminants. This approach is derived from the American Conference of Governmental Industrial Hygienists guidelines for setting threshold limit values for contaminant mixtures. Since the effects on humans of many atmospheric contaminants are considered to be additive, this mixture approach is applied to 16 contaminant groups. The groups considered in the T-value calculation used in the TCCS computer program are the following:

1. Alcohols
2. Aldehydes
3. Aromatic hydrocarbons
4. Esters
5. Ethers
6. Chlorocarbons
7. Chlorofluorocarbons
8. Fluorocarbons
9. Hydrocarbons
10. Inorganic acids
11. Ketones
12. Mercaptans and sulfides
13. Nitrogen oxides
14. Organic acids
15. Organic nitrogens
16. Miscellaneous

The group numbers used in the computer program output correspond to the above group listing.

The T-value is calculated for each group by calculating the sum of the ratios of the contaminants' concentrations to their maximum allowable concentration, while the overall T-value is the sum of the group T-values for the alcohols, aldehydes, aromatic hydrocarbons, esters, ethers, hydrocarbons, inorganic acids, ketones, nitrogen oxides, organic acids, and miscellaneous groups. These calculations are conducted according to the following equations:

$$T_{\text{group}} = \Sigma\, C_c/C_m\,, \tag{B1}$$

$$T_{\text{overall}} = \Sigma\, T_{\text{group}}\,, \tag{B2}$$

where C_c is the contaminant concentration in the atmosphere in mg/m^3 and C_m is the maximum allowable concentration in the atmosphere in mg/m^3.

The criteria for acceptability are the following:

1. The T-value for each group must be less than one

2. The overall T-value must be less than one.

If either of these criteria are exceeded, the atmosphere is considered unacceptable.

REFERENCES

1. Perry, J.L.: "Computerized Atmospheric Trace Contaminant Control Simulation for Manned Spacecraft." NASA TM–108409, NASA, George C. Marshall Space Flight Center, Marshall Space Flight Center, AL, June 1993.

2. Leban, M.I.: "Modification of Trace Contaminant Computer Program." Engineering Memo TCC-0077, Lockheed Missiles and Space Co., Sunnyvale, CA, December 14, 1993.

3. "Preliminary Documentation for Space Station Gaseous Contaminant Control Program", NASA contract NAS8-36406, Bioastronautics Division, Lockheed Missiles and Space Co., Inc., Sunnyvale, CA, September 5, 1986, pp. 33–61.

4. Olcott, T.M.: "Development of a Sorber Trace Contaminant Control System Including Pre- and Post-sorbers for a Catalytic Oxidizer." NASA CR–2027, Lockheed Missiles and Space Co., Sunnyvale, CA, May 1972, p. 26.

5. Ibid., p. 24

6. Robell, A.J., Arnold, C.R., Wheeler, A., Kersels, G.J., and Merrill, R.P.: "Trace Contaminant Adsorption and Sorbent Regeneration." NASA CR–1582. Lockheed Missiles and Space Co., Palo Alto, CA, September 1970.

7. Olcott, T.M.: "Development of a Sorber Trace Contaminant Control System Including Pre- and Post-sorbers for a Catalytic Oxidizer." NASA CR–2027, Lockheed Missiles and Space Co., Sunnyvale, CA, May 1972.

8. Ibid., p. 31.

9. "Preliminary Documentation for Space Station Gaseous Contaminant Control Program", NASA contract NAS8-36406, Bioastronautics Division, Lockheed Missiles and Space Co., Inc., Sunnyvale, CA, September 5, 1986, p. 59.

10. Jagow, R.B., Katan, T., Lamparter, R.A., Dibble, K.P., Masi, O.H., Richardi, C.L., Kawasaki, E.H.: "Investigation and Testing of Low Temperature Carbon Monoxide Catalysts." LMSC-D556669, NASA contract NAS 8–31960, Lockheed Missiles and Space Co., Inc., Sunnyvale, CA, January 20, 1977, p. 29.

11. Kissinger, L.D., and Verostko, C.E.: "Inherent Ammonia Removal Potential of the Space Shuttle Orbiter Condensing Heat Exchanger." JSC-08797, NASA Lyndon B. Johnson Space Center Crew Systems Division, Houston, TX, August 23, 1976, p. 10.

12. Ibid., p.14.

APPROVAL

TRACE CONTAMINANT CONTROL SIMULATION COMPUTER PROGRAM—VERSION 8.1

By J.L. Perry

The information in this report has been reviewed for technical content. Review of any information concerning Department of Defense or nuclear energy activities or programs has been made by the MSFC Security Classification Officer. This report, in its entirety, has been determined to be unclassified.

J.C. BLAIR
Director, Structures and Dynamics Laboratory

☆ U.S. GOVERNMENT PRINTING OFFICE 1994—533-108/00083